НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

КАФЕДРА СИСТЕМНОГО ПРОГРАМУВАННЯ І
СПЕЦІАЛІЗОВАНИХ КОМП'ЮТЕРНИХ СИСТЕМ

«На правах рукопису»
УДК 004.05

«До захисту допущено»

Завідувач кафедри СПСКС

__________ В.П.Тарасенко_
(підпис) (ініціали, прізвище)

"___"_____________2018р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності 123 Комп'ютерна інженерія
(Спеціалізовані комп'ютерні системи)

на тему: «Метод обчислення надійності відмовостійкої багатопроцесорної системи, що складається з декількох підсистем»

Виконав: студент II курсу, групи КВ-63м

Корнєйчик Борис Анатолійович __________
(підпис)

Науковий керівник доц. каф. СП СКС Романкевич В.О. __________
(підпис)

Рецензент ст. вик. ПМА Мальчиков В.В. __________
(підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____________
(підпис)

Київ – 2018 року

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Факультет прикладної математики

Кафедра системного програмування і спеціалізованих комп’ютерних систем

Рівень вищої освіти – другий (магістерський)
Спеціальність 123 Комп‘ютерна інженерія
(Спеціалізовані комп‘ютерні системи)

ЗАТВЕРДЖУЮ
Завідувач кафедри СПСКС
__________ _В.П.Тарасенко__
(підпис) (ініціали, прізвище)

«___»_____________2018р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Корнєйчику Борису Анатолійовичу

1. Тема дисертації: «Метод обчислення надійності відмовостійкої багатопроцесорної системи, що складається з декількох підсистем»,
науковий керівник дисертації доц. каф. СП СКС Романкевич В.О.,
затверджені наказом по університету від «22» березня 2018 р. №986-с

2. Термін подання студентом дисертації 11 травня 2018 р.

3. Об’єкт дослідження: багатопроцесорні відмовостійкі системи.

4. Предмет дослідження: методи та засоби розрахунку показників надійності відмовостійких багатопроцесорних систем шляхом виконання статистичних експериментів з моделями їх поведінки в потоці відмов.

5. Перелік завдань, які потрібно розробити: на основі відомих методів розрахунку надійності цифрової апаратури і статистичних методів створити методики розрахунку надійності відмовостійкої багатопроцесорної системи, що складається з декількох підсистем.

6. Перелік ілюстративного матеріалу: 1) Побудова GL-моделей для підсистем відмовостійкої багатомодульної системи; 2) Формули для розрахунку надійності ВБС статистичним методом з розбивкою вихідної системи на підсистеми; 3) Алгоритм Рушді; 4) Загальний алгоритм розрахунку надійності системи; 5) Алгоритм Барлоу та Хейдтманна; 6) Початковий етап розрахунку надійності (функція повного перебору Відмовних ситуацій) для системи в цілому і для підсистеми; 7) Формули для Розрахунку надійності ВБС статистичним методом без розбивки вихідної системи на підсистеми.

7. Перелік публікацій: ПМК 2018 «Алгоритм перетворення GL-моделей»

8. Дата видачі завдання 5 вересня 2016 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Вивчення літератури за тематикою дисертації		
2	Розроблення та узгодження технічного завдання		
3	Аналіз існуючих рішень		
4	Підготовка матеріалів першого розділу магістерської дисертації		
5	Підготовка матеріалів другого розділу магістерської дисертації		
6	Підготовка матеріалів третього розділу магістерської дисертації		
7	Підготовка матеріалів четвертого розділу магістерської дисертації		
8	Оформлення документації магістерської дисертації		
9	Перевірка на плагіат		
10	Попередній розгляд магістерської дисертації на кафедрі		

Студент ______________ ________________________
(підпис) (ініціали, прізвище)

Науковий керівник дисертації ______________ ________________________
(підпис) (ініціали, прізвище)

РЕФЕРАТ

Актуальність теми обумовлена збільшенням обсягів оброблюваної інформації, вимогам до швидкості її обробки і до надійності апаратних і програмних засобів, що використовуються для цих цілей. У зв'язку з цими обставинами системи обробки інформації і особливо системи управління проектуються відмовостійкими, багатопроцесорними і реконфігурованими, а сфера використання відмовостійких багатопроцесорних обчислювальних систем і систем управління все більше і більше розширюється. Найбільш широке поширення такі системи отримують у тих областях, де відмови і збої апаратури можуть привести до великих фінансових збитків, екологічних катастроф і людських жертв (банківські системи, атомні електростанції, системи управління літаків і ракет, медичні системи та ін).

При розробці відмовостійких багатопроцесорних систем (ВБС) дуже важливо вибрати найбільш оптимальний з точки зору надійності і вартості варіант структури ще на етапах проектування та визначити його так звані «вузькі» місця. Для цього необхідно створювати математичні моделі, що адекватно відбивають поведінку ВБС у потоці відмов і розробляти методи оцінки параметрів надійності ВБС на основі цих моделей, що дозволяють отримати результат обчислень в прийнятний час доступними обчислювальними засобами.

Об'єктом дослідження є багатопроцесорні відмовостійкі системи.

Предметом дослідження є методи та засоби розрахунку показників надійності відмовостійких багатопроцесорних систем шляхом виконання статистичних експериментів з моделями їх поведінки в потоці відмов.

Мета роботи. На основі відомих методів розрахунку надійності цифрової апаратури і статистичних методів створення методики розрахунку надійності відмовостійкої багатопроцесорної системи, що складається з декількох підсистем.

Методи дослідження. В роботі використовуються методи теорії ймовірностей і математичної статистики, теорії графів, дискретної математики, а також методи теорії надійності засобів обчислювальної техніки і систем управління.

Наукова новизна полягає в розробці методики оцінки показників надійності відмовостійких багатопроцесорних систем, з використанням розбиття таких систем на підсистеми, обчислення показників надійності цих підсистем за допомогою статистичних експериментів з моделями їх поведінки в потоці відмов і далі обчислення на підставі цих показників надійностних характеристик системи в цілому.

Практична цінність отриманих в роботі результатів полягає в тому, що вони дозволяють оцінити показники надійності всієї ВБС на підставі оцінок надійності її підсистем, що мають меншу складність, і мінімізувати таким чином, обчислювальні та часові витрати, тобто підвищити ефективність розрахунку.

Апробація роботи. Матеріали роботи були оприлюднені на щорічній науково-технічній конференції магістрантів кафедри СКС ФПМ НТУУ "КПІ" ПМК-2018 і на щорічній МНПК "Перспективні системи управління на залізничному, промисловому і міському транспорті".

Структура і обсяг роботи.

У **вступі** обґрунтовується актуальність теми роботи, формулюється мета і завдання дослідження.

У **розділі 1** наведено загальні відомості про ВБС, описано основні поняття і визначення теорії надійності та теорії живучості обчислювальних систем.

У **розділі 2** описано алгоритми побудови графо-логічних моделей (GL-моделей) циклічного типу базових ВБС, стійких до m-кратних відмов для випадків $m = 2$ і $m > 2$. Для випадку $m > 2$ наведено загальний підхід до побудови GL-моделі і оптимізований алгоритм побудови GL-моделі.

У **розділі 3** розглянуті відомі методи розрахунку надійності базових ВБС (так званих k-out-of-n) систем.

У **розділі 4** розглянуто метод розрахунку надійності ВБС, заснований на використанні GL-моделей та методу статистичного розрахунку надійності, без розділения вихідної системи на підсистеми. Запропоновано метод розрахунку надійності ВБС, який також базується на використанні GL-моделей і виконанні статистичного експерименту, але з розбиттям вихідної системи на підсистеми.

У **висновках** зроблено загальні висновки по роботі, проаналізовано отримані результати.

У **додатку 1** наведено копії графічного матеріалу.

Атестаційна магістерська робота представлена на 90 сторінках і складається з введення, 4 розділів, висновку і містить 6 рисунків, 15 таблиць, список використаних джерел з 24 найменувань.

Ключові слова: БАГАТОПРОЦЕСОРНІ ВІДМОВОСТІЙКІ СИСТЕМИ, НАДІЙНІСТЬ, ГРАФО-ЛОГІЧНА МОДЕЛЬ.

РЕФЕРАТ

Актуальность темы обусловлена увеличением объемов обрабатываемой информации, требованиям к скорости ее обработки и к надежности аппаратных и программных средств, используемых для этих целей. В связи с этими обстоятельствами системы обработки информации и особенно системы управления проектируются отказоустойчивыми, многопроцессорные и реконфигурируемые, а сфера использования отказоустойчивых многопроцессорных вычислительных систем и систем управления все больше и больше расширяется. Наиболее широкое распространение такие системы получают в тех областях, где отказы и сбои аппаратуры могут привести к большим финансовым убыткам, экологических катастроф и человеческих жертв (банковские системы, атомные электростанции, системы управления самолетов и ракет, медицинские системы и др.)

При разработке отказоустойчивых многопроцессорных систем (ВБС) очень важно выбрать наиболее оптимальный с точки зрения надежности и стоимости вариант структуры еще на этапах проектирования и определить его так называемые «узкие» места. Для этого необходимо создавать математические модели, адекватно отражают поведение ВБС в потоке отказов и разрабатывать методы оценки параметров надежности ВБС на основе этих моделей, позволяющих получить результат вычислений в приемлемое время доступными вычислительными средствами.

Объектом исследования является многопроцессорные отказоустойчивые системы.

Предметом исследования являются методы и средства расчета показателей надежности отказоустойчивых многопроцессорных систем путем выполнения статистических экспериментов с моделями их поведения в потоке отказов.

Цель работы. На основе известных методов расчета надежности цифровой аппаратуры и статистических методов создания методики расчета надежности отказоустойчивой многопроцессорной системы, состоящей из нескольких подсистем.

Методы исследования. В работе используются методы теории вероятностей и математической статистики, теории графов, дискретной математики, а также методы теории надежности средств вычислительной техники и систем управления.

Научная новизна заключается в разработке методики оценки показателей надежности отказоустойчивых многопроцессорных систем, с использованием разбиения таких систем на подсистемы, вычисления показателей надежности этих подсистем с помощью статистических экспериментов с моделями их поведения в потоке отказов и дальше вычисления на основании этих показателей надежностных характеристик системы в целом.

Практическая ценность полученных в работе результатов заключается в том, что они позволяют оценить показатели надежности всей ВБС на основании оценок надежности ее подсистем, имеющих меньшую сложность, и минимизировать таким образом, вычислительные и временные затраты, то есть повысить эффективность расчета.

Апробация работы. Материалы работы были обнародованы на ежегодной научно-технической конференции магистрантов кафедры СКС ФПМ НТУУ "КПИ" ПМК-2018 и на ежегодной МНПК "Перспективные системы управления на железнодорожном, промышленном и городском транспорте".

Структура и объем работы.

Во введении обосновывается актуальность темы работы, формулируется цель и задачи исследования.

В разделе 1 приведены общие сведения о ВБС, описаны основные понятия и определения теории надежности и теории живучести вычислительных систем.

В разделе 2 описано алгоритмы построения графо-логических моделей (GL-моделей) циклического типа базовых ВБС, устойчивых к m-кратных отказов для случаев m = 2 и m> 2. случае m> 2 приведен общий подход к построению GL-модели и оптимизирован алгоритм построения GL-модели.

В разделе 3 рассмотрены известные методы расчета надежности базовых ВБС (так называемых k-out-of-n) систем.

В разделе 4 рассмотрен метод расчета надежности ВБС, основанный на использовании GL-моделей и метода статистического расчета надежности, без роздиления исходной системы на подсистемы. Предложен метод расчета надежности ВБС, который также базируется на использовании GL-моделей и выполнении статистического эксперимента, но с разбивкой исходной системы на подсистемы.

В выводах сделаны общие выводы по работе, проанализированы полученные результаты.

В **приложении 1** приведены копии графического материала.

Аттестационная магистерская работа представлена на 90 страницах и состоит из введения, 4 глав, заключения и содержит 6 рисунков, 15 таблиц, список использованных источников из 24 наименований.

Ключевые слова: многопроцессорных отказоустойчивых СИСТЕМЫ, НАДЕЖНОСТЬ, граф-логической модели.

Abstract

The urgency of the topic is due to an increase in the volumes of processed information, the requirements for the speed of its processing and the reliability of hardware and software used for these purposes. In connection with these circumstances, the information processing system and especially the control system are designed fault-tolerant, multiprocessor and reconfigurable, and the scope of use of fault-tolerant multiprocessor computer systems and control systems is increasingly expanding. Such systems are most widespread in those areas where failures and equipment failures can lead to large financial losses, environmental disasters and human casualties (banking systems, nuclear power plants, air and rocket systems, medical systems, etc.).

When designing fault-tolerant multiprocessor systems (VSPs) it is very important to choose the most optimal in terms of reliability and cost of the design option at the design stages and identify its so-called "bottlenecks". To do this, you need to create mathematical models that adequately reflect the behavior of the VBS in the flow of failures and develop methods for assessing the reliability of the VBS based on these models, which allow obtaining the result of the calculations in an affordable time available computing means.

The object of research is multiprocessor fault tolerance systems.

The subject of the study is methods and tools for calculating the reliability of fail-safe multiprocessor systems by performing statistical experiments with models of their behavior in the flow of failures.

The purpose of the work. On the basis of known methods for calculating the reliability of digital apparatus and statistical methods for creating a method for calculating the reliability of fault-tolerant multiprocessor system, consisting of several subsystems.

Research methods. Methods of probability theory and mathematical statistics, theory of graphs, discrete mathematics, as well as methods of reliability theory of computer facilities and control systems are used in this work.

The scientific novelty consists in the development of a method for evaluating the reliability of fail-safe multiprocessor systems, using the partition of such systems on a subsystem, calculating the reliability indices of these subsystems by means of statistical experiments with models of their behavior in the flow of failures and further calculating on the basis of these indices the reliability characteristics of the system as a whole.

The practical value of the results obtained in the work is that they allow us to evaluate the reliability of the entire VBS based on the reliability of its subsystems with less complexity and thus minimize computational and time costs, that is, to improve the efficiency of the calculation.

Test work. The materials of the work were announced at the annual scientific and technical conference of the undergraduate students of the faculty SCS FPM NTUU "KPI" PMK-2018 and at the annual MNPK "Perspective systems of management in rail, industrial and urban transport".

Structure and scope of work.

The introduction substantiates the relevance of the topic of work, formulates the purpose and objectives of the study.

Section 1 provides general information about the VBS, describes the basic concepts and defines the theory of reliability and the theory of the survivability of computing systems.

Section 2 describes the algorithms for constructing graph-logic models (GL-models) of cyclic type of basic GMSs, resistant to m-fold failures for cases m = 2 and m> 2. For case m> 2, we give a general approach to the construction of the GL model and optimized algorithm for constructing GL-model.

Section 3 discusses known methods for calculating the reliability of basic KS (so-called k-out-of-n) systems.

Section 4 discusses the method of calculating the reliability of the VBS, based on the use of GL-models and the method of statistical reliability calculation, without separating the original system on the subsystem. A method for calculating the reliability of the VBS is also proposed, which is also based on the use of GL-models and the performance of a statistical experiment, but with the partition of the original system on the subsystem.

The conclusions are general conclusions on the work, analyzed the results.

Appendix 1 provides copies of graphic material.

The attestation master's work is presented on 90 pages and consists of an introduction, 4 chapters, a conclusion and contains 6 figures, 15 tables, list of the used sources from 24 titles.

Keywords: MANUAL PROCESSOR DISORDER SYSTEMS, RELIABILITY, GRAPHIC LOGICAL MODEL.

Зміст

Список скорочень

ВБС – відмовостійка багатопроцессорна система

ОС – обчислювальна система

ІОС – информаційно-обчислювальна система

ТЗ – технічне завдання

ПО – программное забезпечення

GL-модель – графо-логічна модель

СОГ – сигнальний орієнований граф

Метод СНП – метод суми непересічних добутків

Метод ВВ – метод включення і виключення

ВСТУП

В сучасному світі постійно розширюється область застосування багатопроцесорних обчислювальних систем. Завдяки цьому багатопроцесорні системи набувають все більшого поширення. В основному – в системах управління складними відповідальними об'єктами, такими як літаки, ракети, космічні об'єкти, різні технологічні процеси, фінансові розрахунки, наземний транспорт, системи забезпечення життєдіяльності в медицині та ін. Зростає і складність цих систем.

Природними є підвищені вимоги, що пред'являються до надійнісних характеристик багатопроцесорних ОС, так як навіть незначні порушення обчислювального процесу в них можуть призвести до аварійних або фатальних ситуацій або до таких наслідків, як порушення технологічного процесу, помилки у фінансових розрахунках і т. д. Для того, щоб ОС відповідала цим вимогам, необхідно забезпечувати її надійність на всіх етапах розробки, виробництва і експлуатації, використовувати різні методи підвищення надійності.

Одним із методів підвищення надійності ОС є введення відмовостійкості. Під відмовостійкістю інформаційно-обчислювальної системи (ІОС) зазвичай розуміється така властивість, яка дозволяє цій системі продовжувати обчислювальний процес після виникненні несправності. В даний час для забезпечення відмовостійкості широко використовується введення апаратурної та інформаційної надмірностей. Але обов'язково потрібно враховувати той факт, що введення апаратурної надмірності для забезпечення відмовостійкості системи дуже часто спричиняє за собою зниження надійності всієї системи, зумовлене збільшенням кількості її компонентів. Крім того, продуктивність системи також часто знижується за рахунок часу, який витрачається на виконання всіх необхідних діагностичних процедур.

Відмовостійкість ОС може забезпечуватися активною або пассивною схемою.

Пасивна відмовостійкість полягає в тому, що у системи є властивість – не втратити свої функціональні якості (можливості) у разі відмови окремих елементів. Іноді кажуть, що відмова маскується системою.

Приклад пасивно відмовостійких систем – системи з мажоритарних органом.

Забезпечення відмовостійкості за активною схемою (активної відмовостійкості) грунтується на виділяємих процессах виявлення відмови, локалізації відмови і реконфігурації системи.

В даний час багатопроцесорні системи та системи управління дуже часто проектуються із забезпеченням відмовостійкості за активною схемою у зв'язку з пред'явленням підвищених вимог до їх надійнісних характеристик. Такі системи мають аппаратурную і часову надмірність, деякі їх компоненти дублюються або мажоруються. Вони є самотестируваними, так як у будь-який момент часу система повинна знати стан всіх своїх модулів (справний або несправний), щоб мати можливість виконати необхідну реконфігурацію і ізоляцію несправних модулів. Інші частини беруть на себе всі функції елементів, що відмовили, або тільки їх основні функції. У зв'язку з цим на сучасному етапі розвитку обчислювальної техніки особливе значення набуває розвиток теорії самотестируваних, реконфігурованих відмовостійких багатопроцесорних систем (ВБС) і дослідження їх властивостей, у тому числі і надійності, методами математичного моделювання.

ВБС зазвичай називають відмовостійку багатопроцесорну систему, здатну до реконфігурації при відмові її модулів – процесорів.

ВБС, стійка до відмов певної кратності, називається базовою. Це означає, що число відмов елементів-процесорів такої системи не повинно перевищувати деяку фіксовану величину, інакше ця система буде нездатною функціонувати з тими ж параметрами.

Для розрахунку параметрів надійності ВБС необхідно моделювати її поведінку в потоці відмов. Це – надзвичайно складна задача, для її виконання потрібнен величезний обсяг обчислень. Існують різні за соєю складністю і точністю методи розрахунку надійності. В основі будь-якого з цих методів лежить та чи інша математична модель, експерименти над якою дозволяють отримати оцінку кількісних параметрів системи. Також, за допомогою математичної моделі, можна визначити реакцію системи на появу певних ситуацій з відмовою компонентів. Від математичної моделі залежить точність і складність конкретного методу розрахунку надійності

В даній роботі для моделювання поведінки ВБС у потоці відмов використовуються графо-логічні моделі ВБС або GL-моделі. Їх перевага полягає в тому, що вони дозволяють проводити аналіз поведінки складних ВБС, що складаються з сотень процесорів доступними обчислювальними засобами, за рахунок компактного представлення критерію, згідно з яким здійснюється оцінка поведінки системи в конкретно заданих умовах. Ці моделі поєднують використання властивостей булевих функцій і графів, причому за рахунок перенесення складності на булеві функції граф, як правило, має досить просту структуру.

Основною метою даної роботи була розробка методу оцінки надійності відмовостійких багатопроцесорних систем, заснованого на використанні статистичного експерименту з GL-моделями, який дозволив би, не перебираючи повні вектора стану системи, отримати результат. Причому щоб статистичний експеримент проводIOCя не з GL-моделлю всієї системи, а з GL-моделями підсистем, що дасть можливість доступними обчислювальними засобами розраховувати надійність систем з великим числом елементів.

1 ОСНОВНІ ХАРАКТЕРИСТИКИ НАДІЙНОСТІ ОБЧИСЛЮВАЛЬНИХ СИСТЕМ.

1.1 Поняття «надійність обчислювальних систем»

У зв'язку з ростом складності сучасних багатопроцесорних обчислювальних систем і систем управління складними і відповідальними об'єктами дуже важливою стає завдання забезпечення достатнього рівня їх надійності.

Надійність обчислювальної системи (ОС) визначається, з одного боку, відсутністю відмов, збоїв і помилок у її роботі, а з іншого – можливістю швидкого відновлення апаратури і обчислювального процесу.

Надійність – складна властивість, яка в залежності від призначення об'єкта та умов його застосування складається з поєднань властивостей: безвідмовності, ремонтопридатності, довговічності і зберігання (ГОСТ 27.002–83).

Основними поняттями теорії надійності є поняття, наведені нижче.

Безвідмовність – властивість об'єкта безупинно зберігати працездатний стан протягом деякого часу або деякого напрацювання.

Напрацювання – обсяг роботи об'єкта (як правило, під напрацюванням розуміють час роботи).

Відмова – подія, що полягає в порушенні працездатності об'єкта. Як правило, відмова викликана фізичним руйнуванням елемента ЕОМ або поступовим порушенням його характеристик. Відмови бувають стійкими і переміжними, залежними і незалежними, раптовими і поступовими. Раптові відмови з'являються в результаті різкої (стрибкоподібної) зміни основних параметрів системи або її елемента. Поступові відмови проявляються в результаті повільногї зміни параметрів системи та їх виходу з області допустимих значень. Оскільки відмови, в загальному випадку, є випадковими подіями, то вони можуть бути залежними і незалежними.

Збій – короткочасне порушення правильної роботи обчислювального пристрою або його елемента, після якого його працездатність самовідновлюється або відновлюється оператором без проведення ремонту (ГОСТ 19542-83).

При розрахунку надійності систему прийнято поділяти на елементи.

Елемент розрахунку надійності – це частина системи, призначена для виконання заданих функцій, яка має свій узагальнений кількісний показник надійності.

Система – сукупність спільно діючих об'єктів, призначених для самостійного виконання заданих функцій. Системи поділяються на відновлювані та невідновлювані. Відновлювані системи – це системи, які в разі відмови можуть бути відновлені. Відповідно, невідновлювані системи – це ті системи, які після відмови відновленню не підлягають або не піддаються відновленню в процесі експлуатації. В якості прикладу відновлюваної системи можна привести комп'ютер, що працює в стаціонарних умовах, а комп'ютер, що встановлений на борту літака і відмовив у польоті не є відновлюваною системою.

Для обох видів систем вводиться ряд понять:

- напрацювання – тривалість або обсяг роботи системи в заданих умовах;
- напрацювання між відмовами – напрацювання відновлюваної системи між двома відмовами, які виникли послідовно;
- напрацювання на відмову – середнє значення напрацювання відновлюваної системи між відмовами;
- технічний ресурс – сумарне напрацювання системи за період експлуатації до руйнування або до іншого граничного стану;
- термін служби – календарна тривалість експлуатації системи до руйнування або до іншого граничного стану;
- гарантований ресурс - технічний ресурс, яким володіють не менш ніж χ% експлуатованих систем (χ - гарантована ймовірність);

- гарантований термін служби – термін служби, протягом якого виробник гарантує справність системи і несе матеріальну відповідальність за виниклі несправності.

Надійність ОС і її підсистем планується на етапі розробки технічного завдання (ТЗ) і закладається на ранніх етапах розробки – при ескізному проектуванні, забезпечується на наступних етапах розробки – технічному і робочому проектуванні, реалізується в процесі виробництва та підтримується в процесі експлуатації.

Для порівняльної оцінки окремих шляхів забезпечення і підвищення надійності застосовуються розрахункові і експериментальні методи. Значення розрахункових методів більше на перших етапах розробки і знижується на користь експериментального аналізу і перевірки на останніх етапах.

Правильні технічні рішення щодо забезпечення надійності на перших етапах розробки дають значну економію коштів порівняно з випадком, коли прийняті технічні рішення доводиться переглянути на стадіях технічного і робочого проектування або навіть на етапах експлуатації. Тому виняткове значення для економного забезпечення надійності набуває правильне розуміння і застосування розрахункових методів її оцінки для обґрунтованого вибору найкращих технічних рішень з точки зору надійності.

Тобто, забезпечувати надійність ОС необхідно на всіх етапах її життєвого циклу. Виділяють наступні етапи життєвого циклу ПС:

1. Створення ТЗ
2. Розробка ескізного проекту ОС
3. Розробка технічного і робочого проектів
4. Виробництво ОС
5. Експлуатація ОС

На кожному з цих етапів надійність ОС забезпечується по-різному. Так, при складанні ТЗ збираються всі наявні дані про аналоги, умови застосування ОС і пред'являються вимоги до функцій, які вона повинна виконувати. За допомогою сукупності цих даних розробляються обґрунтовані вимоги до надійності даної системи.

На етапі ескізного проектування вибирається елементна база, визначаються особливості структури, архітектури та організації розробки системи. За цими даними проводиться попередній розрахунок надійності, виявляються найменш надійні підсистеми і приймаються на цій основі рішення про резервування відповідних підсистем, організації технічного обслуговування (профілактичних і ремонтних робіт). Також досліджується і вирішується питання про доцільність та способи реалізації методів автоматичного відновлення та відмовостійкості в системі.

На етапі технічного і робочого проектування ОС перевіряються і уточнюються раніше прийняті технічні рішення. Основою для цього служать уточнені дані про надійність, отримані на основі розрахунків з урахуванням режимів роботи і точної номенклатури елементів системи, а також результати експериментів над моделями, макетами, експерементальними та промисловими зразками.

Розробляється програмне забезпечення (ПЗ) системи і проводиться його всебічна перевірка за тестами і шляхом імітаційного моделювання на моделі розроблюваної системи.

Практично дуже важливим для забезпечення надійності є виявлення та виправлення всіх помилок у розроблювальній технічної документації.

На етапі виробництва ОС основним є технічний контроль. Він охоплює всі стадії виробничого процесу, починаючи від вхідного контролю якості поступаючих матеріалів і комплектуючих виробів, включаючи контроль якості та відповідності технічної документації друкованих плат, блоків, пристроїв, схемних з'єднаннь, конструкції і закінчуючи випробуванням готової продукції. На цьому етапі також виявляються окремі недоліки в розробці, що впливають на надійність системи, приймаються заходи по їх усуненню.

На етапі експлуатації найважливішими факторами, що забезпечують надійну роботу системи, є контроль та забезпечення умов навколишнього середовища, передбачених проектом, достатня кваліфікація та склад обслуговуючого персоналу, організація та проведення технічного обслуговування і ремонтів в передбаченому порядку.

На цьому ж етапі триває збір відомостей про відмови апаратури і ПЗ, які передаються розробникам з метою усунення причин відмов і уточнення даних для розрахунку надійності [16].

Надійність програмного забезпечення, що розробляється на етапі технічного і робочого проектування ОС, є окремою галуззю досліджень. Системний підхід вимагає розглядати надійність роботи обчислювальної апаратури спільно з ПЗ як надійність обчислювально процесу.

Надійність програмного забезпечення можна визначити як властивість програми, що виражається у виконанні заданих функцій в заданих умовах роботи і на заданій обчислювальній машині. Тобто, це поняття визначається аналогічно до того, як визначається поняття надійності апаратури. Але механізми виникнення відмови апаратури і відмови ПЗ істотно відрізняються один від одного. Якщо відмова апаратури зумовлена, як правило, руйнуванням якихось її елементів, то відмова ПЗ – невідповідністю даного ПЗ поставленим завданням. Ця невідповідність може виникати з двох причин: або розробниками програми було допущено порушення специфікації – технічних вимог до програми, або специфікація неточна або неповна.

Невідповідність з першої причини зустрічається, в першу чергу, в складних програмних системах, де окремі помилки програміста важко доступні для огляду і можуть до певного часу залишатися нерозкритими.

Невідповідність з другої причини виникає, в першу чергу, тому що при складанні специфікації багато факторів, що впливають на роботу програми, невідомі. Вони виявляються тільки поступово, в ході експлуатації програми. Особливо це відноситься до керуючих програм. Крім того, ні в технічних вимогах, ні при перевірці програми неможливо обговорити й перевірити всі ситуації, які виникають при експлуатації цієї програми.

Характерною особливістю помилок, які обумовлюють відмови, є їх скритність, тобто прояв при окремих комбінаціях з величезної кількості можливих комбінацій вихідних даних. Тому такі помилки виявляються не відразу, а тільки в ході тривалої експлуатації. Помилки, що виявляються при будь-яких комбінаціях вихідних даних не небезпечні, оскільки виявляються відразу ж, при перших пробних прогонах програми.

Для прогнозування надійності програм в ході їх експлуатації існують математичні моделі надійності програм, засновані на різних припущеннях про інтенсивність прояву помилок програми.

Можна виділити такі основні способи підвищення та забезпечення надійності програм як удосконалення технології програмування; вибір алгоритмів, не чутливих до різного роду порушень обчислювального процесу (використання алгоритмічної надмірності); резервування програм; контроль і тестування програм з наступною їх корекцією. Перші два способи належать до т. зв. технологічним заходам забезпечення надійності програм, надійність програм, яка може бути досягнута їх застосуванням, обмежена. Тому необхідно використовувати й інші способи підвищення надійності програм. Наприклад, це може бути резервування програмного забезпечення. Але, в той же час, необхідно відзначити, що основним вживаним способом, що дозволяє забезпечити надійність, є тестування програм [16].

1.2 Основні показники надійності

На практиці застосовуються різні показники надійності, що характеризують комплексну властивість «надійність» з різних сторін. Оскільки час до відмови, час між двома відмовами, а також час відновлення – випадкові величини, показники надійності є імовірнісними показниками. Основні показники надійності ОС наведені нижче.

Основною якісною характеристикою надійності є т. зв. функція надійності, або, скорочено, надійність. За визначенням надійність дорівнює ймовірності безвідмовної роботи системи протягом заданого часу або ймовірності того, що при заданих умовах у межах заданої тривалості роботи (напрацювання) системи відмови не виникне, тобто,

$$P(t) = P(T > t), \tag{1.1}$$

де T – час безвідмовної роботи системи,

t – заданий час (період часу, що розглядається),

P(A) – ймовірність події А (у данному випадку А полягає в тому, що $T > t$).

Доповнення ймовірності безвідмовної роботи до одиниці

$$Q(t) = 1 - P(t) \tag{1.2}$$

називається ймовірністю відмови або функцією ненадійності системи, або, скорочено, просто ненадійністю. Ненадійність – ймовірність того, що випадковий час до відмови менше заданого часу t. Тому функція $Q(t)$ збігається з функцією розподілу часу до відмови.

Показники мають наступні властивості, які очевидні з його визначення:

1. $P(0) = 1$, $Q(0) = 0$ (передбачається, що до початку роботи об'єкт (система) є безумовно працездатним);
2. $\lim\limits_{t\to\infty} P(t) = 0$, $\lim\limits_{t\to\infty} Q(t) = 1$ (передбачається, що розглядаються системи, час роботи яких скінченний);
3. $\dfrac{dP(t)}{dt} \le 0$ (функція надійності являеться спадною функцією)
4. $0 \le P(t) \le 1$, $0 \le Q(t) \le 1$
5. $P(t_2) \le P(t_1)$, $Q(t_2) \ge Q(t_2)$, якщо $t_2 \ge t_1$

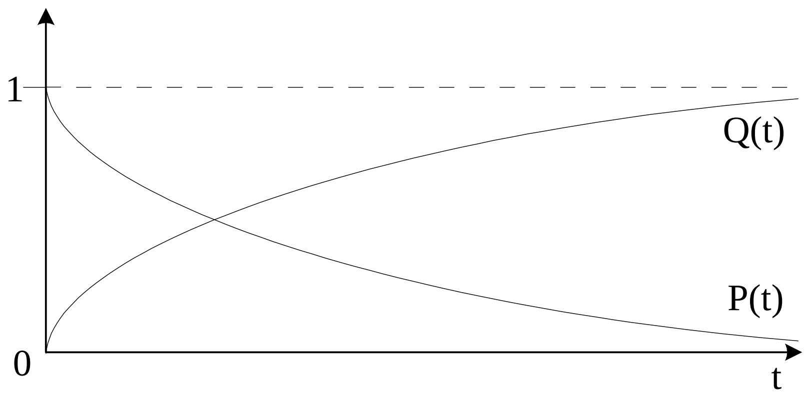


Рисунок.1.1 – Ймовірність безвідмовної роботи

6. $P(t) + Q(t) = 1$ (слідкує безпосередньо з визначень надійності та ненадійності).

Похідна від функції ненадійності називається щільністю розподілу часу безвідмовної роботи

$$f(t) = \frac{dQ}{dt} = -\frac{dP}{dt}. \quad (1.3)$$

В якості показника надійності незручно використовувати функціональну залежність, наприклад, функцію P(t). Тому в технічних умовах зазвичай обумовлюють окремі ординати (одну або дві) функції P(t) при значеннях t, обираних з нормованого ряду t = 100; 500; 1000; 2000; 5000; 10000 годин [16-18].

За статистичним експериментом можна визначити P(t) лише наближено, у вигляді статистичної оцінки, позначається тільдою:

$$\hat{P}(t) = \frac{N(t)}{N_0}, \quad (1.4)$$

де N(t) – кількість безвідмовно працюючих до моменту часу t об'єктів при початковій кількості N_0.

Точки розриву функції $\hat{P}(t)$ відповідають окремим відмовам, тому вигляд $\hat{P}(t)$ змінюється випадково, від експерименту до експерименту навіть при незмінній функції $P(t)$. При необмеженому збільшенні N_0 функція наближається до функції $P(t)$.

Середньоквадратична похибка оцінки (1.4) визначається на основі теореми Муавра – Лапласа виразом

$$\sigma_{\hat{P}} \approx \sqrt{\frac{P(1-P)}{N_0}},$$

де – значення $P(t)$ для розглянутого моменту часу. Як випливає з цього виразу, для забезпечення малої похибки потрібно проводити експеримент над великою кількістю зразків.

Інтенсивність відмов $\lambda(t)$ виражає інтенсивність процесу виникнення відмов (визначається як середнє число відмов в одиницю часу). Ця характеристика надійності є зручною для розрахунків.

З плином часу кількість випробовуваних зразків в наслідку їх відмов зменшується, тому однакові значення щільності розподілу $f(t)$ говорять про тим більшу інтенсивность процесу виникнення відмов, чим менше ймовірність безвідмовної роботи (тобто кількість зразків). Тому мірою інтенсивності відмов слугує відношення

$$\lambda(t) = \frac{f(t)}{P(t)} = \frac{\left(\frac{-dP(t)}{dt}\right)}{P(t)} \qquad (1.5)$$

Вирішивши відношення (1.5) як диференціальне рівняння, отримаємо наступну формулу:

$$P(t) = e^{-\int_0^t \lambda(t)dt}. \qquad (1.6)$$

Статистично інтенсивність потоку відмов оцінюється за відношенням

$$\lambda_c = \frac{N(t, t+\Delta t)}{N(t)\Delta t},$$

як число відмов $N(t,\ t{+}\Delta t)$ в достатньо малому інтервалі часу, віднесена до числа тих, що залишилися до моменту часу t зразків N(t) та інтервалу часу Δt [16].

Усі розглянуті вище показники надійності є функціональними показниками, крім них на практиці широко використовуються числові показники надійності. Найважливішим серед них є середній час безвідмовної роботи T_0. Його називають також середнім напрацюванням до відмови і просто напрацюванням до відмови T_0 визначається як математичне сподівання випадкової величини T – час безвідмовної роботи:

$$T_0 = M(T) = \int_0^\infty tf(t)dt = -\int_0^\infty t\frac{dP(t)}{dt}dt = -\int_0^\infty tdP(t) = -(tP(t)\,|_0^\infty + \int_0^\infty P(t)dt) =$$

$$= \int_0^\infty P(t)dt \qquad (1.7)$$

Середній час безвідмовної роботи є природним показником надійності, але віно нічого не говорить про характер розподілу часу до відмови. Наприклад, дві зовсім різні функції $P_1(t)$ і $P_2(t)$, що виражають дві дуже різні ймовірності безвідмовної роботи при малому напрацювання($t << T_0$), можуть характеризуватися однаковими значеннями $T_{0_1} = T_{0_2}$. [16]

Гарантований ресурс t_γ (або гамма - відсоткове напрацювання) – це технічний ресурс, яким володіють не менш ніж γ % експлуатованих систем, де γ – гарантована ймовірність безвідмовної роботи. Вона виражає абсциссу функції надійності при заданій ординаті, тобто визначається як значення функції, оберненої до функції надійності – $P^{-1}(\gamma)$ при заданому значенні аргументу:

$$t_\gamma = P^{-1}(\gamma), \qquad (1.9)$$

де γ – деяка гарантована ймовірність.

Таким чином, гамма - процентне напрацювання t_γ – час, протягом якого гарантується безвідмовна робота об'єкта із заданою ймовірністю γ.

Усі розглянуті вище показники надійності придатні як для невідновлюваних, так і для відновлюваних систем. Однак останні вони характеризують від початку експлуатації до першої відмови.

Але, так як відновлювальна система після відмови ремонтується, після чого її працездатність відновлюється, то цих характеристик недостатньо для повного опису роботи такої системи.

Для більш повного опису відновлюваної системи необхідно крім процесу виникнення відмови описати ще і процес відновлення (ремонту) системи. Для цього використовуються наступні показники.

Функція ремонту – це ймовірність того, що час відновлення системи $T_в$ буде меншим за заданий час t, тобто

$$R(t) = P\{T_в \le t\}, \tag{1.10}$$

де $R(t)$ – випадкова величина.

$R(t)$ є функцією розподілу часу відновлення, вона – аналог функції ненадійності [16-18].

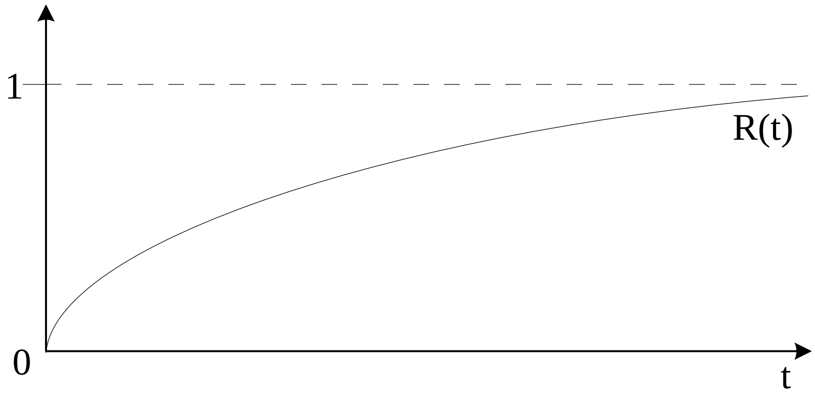


Рисунок. 1.2 – Функція ремонту

За аналогією з щільністю розподілу безвідмовної роботи і з інтенсивністю відмов для відновлюваних систем вводяться поняття щільності часу відновлення та інтенсивність ремонту - $r(t)$ та $\mu(t)$ відповідно.

$$r(t) = \frac{dR(t)}{dt} \tag{1.11}$$

$$\mu(t) = \frac{r(t)}{1 - R(t)} \tag{1.12}$$

Статистично інтенсивність ремонту визначається як

$$\mu_c = \frac{N'_1(t + \Delta t) - N'_1(t)}{N'_1(t) \cdot \Delta t}, \tag{1.13}$$

де $N'_1(t)$ – число відновлених до моменту часу t об’єктів [17].

На основі формул (1.11) і (1.12) за аналогією з $Q(t)$, для $R(t)$ можна записати наступну формулу:

$$R(t) = 1 - e^{-\int_0^t \mu(t)dt} \tag{1.14}$$

Для характеристики працездатності відновлюваної системи в довільний момент часу використовують функцію готовності $G(t)$, яка за визначенням дорівнює ймовірності того, що в момент часу t система справна.

Функція готовності може бути отримана з інших показників надійності відновлюваної системи, наприклад, з функції ремонту $R(t)$ і ймовірності безвідмовної роботи $P(t)$.

Середній час справного стану системи $T_и$ за деякий час її роботи T:

$$T_и = \int_0^T G(t)dt \tag{1.15}$$

Параметр потоку відмов $\omega(t)$ висловлює питому кількість відмов за одиницю часу (на один зразок апаратури). Статистично цей показник оцінюється за наступною формулою:

$$\hat{\omega}(t) = \frac{N(t,\, t + \Delta t)}{N_0 \Delta t}.$$

Середнє напрацювання на відмову T_0 визначається як відношення напрацювання відновлюваного об'єкта до математичного очікування числа його відмов протягом цього часу[16].

Коефіцієнт готовності $К_г$ визначається як ймовірність того, що в довільний момент часу t об'єкт знаходиться в стані працездатності (крім планованих періодів, протягом яких застосування об'єкта за призначенням не передбачається). Цей показник надійності використовується, якщо крім факту відмови необхідно враховувати час відновлення, оскільки він не знехтовно малий.

Статистична оцінка коефіцієнта готовності обчислюється як

$$K_г = \frac{N_в(t)}{N_0}, \tag{1.16}$$

де $N_в(t)$ – число об'єктів, що перебувають у робочому стані в момент часу t.

Різниця $N_в$ – N виражає кількість об'єктів, що знаходяться в момент часу t в стані відновлення (ремонту).

Стаціонарний коефіцієнт готовності $K_{г.ст}$ визначається як межа (1.16) при $t \to \infty$ або як відношення тривалості безвідмовної роботи системи за даний період експлуатації до суми цієї тривалості та тривалості часу відновлення [16-18]:

$$K_{г.ст.} = \frac{T_u}{T}. \tag{1.17}$$

1.3 Поняття і візначення живучості

Здатність системи виконувати встановлений мінімальний обсяг своїх функцій при зовнішніх впливах, не передбачених умовами нормальної експлуатації, протистояти таким впливам, здійснювати вибір оптимального режиму функціонування за рахунок власних внутрішніх ресурсів, перебудови структури, зміни функцій окремих підсистем та їх поведінки характеризує властивість живучості. Ця властивість властива не тільки біологічним, але й складним технічним системам.

Характер поведінки системи вибирається відповідно до змін зовнішніх умов і з функціональним інваріантом системи, який можна назвати внутрішньою метою її функціонування. Вибір поведінки припускає також наявність деякої множини можливих різних наслідків, об'єднаних загальною властивістю відповідності одній зовнішній причині в даних умовах. Отже, здійснювати вибір поведінки можуть тільки системи, що виключають жорсткий зв'язок зовнішньої причини вибору з фактичною поведінкою системи в результаті вибору (зовнішні причини викликають наслідки, які не можуть бути передбачені однозначно).

Живучість може оцінюватися відносно різних рівнів організації системи (функціонального, структурного, елементного, тощо). В залежності від типу і призначення системи живучість забезпечується різними засобами, які включаються в систему при проектуванні, або використовуються внутрішні можливості вже готової системи.

З точки зору якісного виконання функцій системою, живучість характеризує її здатність виконувати задані функції з деякою допустимою якістю, причому впливи на систему можуть мати як природний, так і навмисний характер.

Теорія живучості обчислювальних систем вивчає особливості поведінки систем та їх компонентів при зміні умов функціонування. До факторів, що викликають зміни, можна віднести кліматичні умови, збої апаратури, помилкові дії людини, цілеспрямовані руйнівні дії. Предметом дослідження в теорії живучості є також функціонування окремих компонентів (елементів, підсистем і т. д.) ОС, однак, властивості компонентів вивчаються з метою дослідження поведінки всієї системи, стосовно оцінки живучості, з урахуванням взаємодії між іншими компонентами.

Під живучістю інформаційно-обчислювальної системи (ІОС) мається на увазі властивість системи адаптуватися до нової ситуації і протистояти шкідливим впливам, реалізуючи мету функціонування за рахунок відповідної зміни своєї структури і поведінки. Властивість живучості передбачає здатність системи функціонувати при наявності відмов та їх нагромадженні. Обчислювальна система, що має більше напрацювання на відмову, з точки зору надійності є кращою. Більш живучою є та ОС, яка може відповідати меті функціонування за рахунок компенсації більшого числа відмов. Причому, якщо в відмовостійкій системі за рахунок використання надлишкових ресурсів після відмови відновлення здійснюється за рахунок повернення до колишньої структури та поведінки, то в живучій системі задана ефективність виконання мети функціонування забезпечується за рахунок відповідної зміни структури і поведінки (хоча можливі випадки, коли змін немає).

До особливостей ОС, у яких може проявлятися властивість живучості, необхідно віднести наступні:

- наявність єдиної мети функціонування для всієї системи;

- складність організації - система складається з ряду компонентів (підсистем, комп'ютерів, окремого ряду мікропроцесорів і т.п.);
- багатофункціональність окремих компонентів системи;
- доступність каналів зв'язку для інформаційного обміну між окремими компонентами;
- наявність засобів захисту, контролю, діагностики та самоорганізації.

При аналізі живучості ОС розрізняють структурну і функціональну живучість.

При розгляді структурної живучості враховуються топологія мережі межкомпонентного зв'язку та надійнісні характеристики компонентів. Завдання, пов'язані з аналізом структурної живучості, можна звести до завдань надійності, зв'язності топологічних структур, в залежності від ведення поняття «руйнування».

При дослідженні функціональної живучості ОС особливості топології мережі межкомпонентних зв'язків враховуються опосередковано. Передбачається, що в ОС забезпечується необхідна зв'язність працездатних компонентів.

При аналізі функціональної живучості ОС характеризуються:

- метою функціонування;
- множиною завдань, рішення яких забезпечується в ній $Q = \{q_1, q_2, \ldots, q_m\}$;
- множиною компонентів $\{S_1, S_2, \ldots, S_n\}$, які є складовими частинами системи (в даному контексті їх ще називають функціональними модулями системи).

В процесі роботи системи її функціональні модулі можуть перебувати в одному з наступних станів:

- модуль працездатний;
- модуль непрацездатний;

- модуль частково працездатний, тобто модуль працездатний, але знизилися (у допустимих межах) значення яких-небудь показників якості його функціонування.

Якщо позначити через W деяку множину станів обчислювальної системи, визначені через відмови функціональних модулів системи, то для живучих ОС можна виділити три типи цілей функціонування.

1. Система повинна вирішувати множину завдань $Q = \{q_1, q_2, \ldots, q_m\}$ із заданою ефективністю (наприклад, з заданими продуктивністю, часом рішення) в довільному стані $w \in W$

$$\prod_{i=1}^{m} x(q_i) = 1 \qquad x(q_i) = \begin{cases} 1, \text{ якщо задача здійсненна в ОС} \\ 0, \text{ в іншому випадку.} \end{cases}$$

2. Система повинна забезпечувати вирішення деякої підмножини задач $Q^* \subset Q$ в довільному стані $w \in W$. Завдання з множини $Q \setminus Q^*$ можуть вирішуватися в системі у разі появи такої можливості в деякому стані $w' \in W$, тобто, множина завдань, які виконуються в системі, залежить від стану, в якому знаходиться ВС, і не є постійною, тоді

$$\prod_{q_i \in Q^*} x(q_i) = 1.$$

Саме такий тип функціонування дозволяє розглядати живучість як здатність збереження заданої множини функцій, збереження ваги системи (вага системи визначається як сума реалізованих в ній функцій) і т. п.

3. Система повинна забезпечувати виконання хоча б однієї задачі в довільному стані $w \in W$, т.е. $\sum_{i=1}^{m} x(q_i) \geq 1$.

Оцінка живучості ОС повинна завжди будуватися з урахуванням мети функціонування. Порівняння показників живучості різних ОС може також здійснюватися при однакових цілях функціонування.

Оцінка живучості однієї і тієї ж самої ОС може змінюватися при зміні типу цілей функціонування.

Крім мети функціонування істотний вплив на кількісні показники живучості ОС, надають параметри, що визначають умови працездатності системи.

Мета функціонування – це поняття, яке вводиться для живучих, відмовостійких ОС, але зміна її передбачається можливою лише для ОС, що мають властивість живучості.

Оцінка живучості ОС з заданою конфігурацією, умовами та метою функціонування, множиною розв'язуваних задач належить до завдань аналізу.

При аналізі систем в процесі експлуатації способи оцінки живучості будуються, як правило, на основі вимірювання параметрів функціонування. Оцінки отримують методами, що використовують статистичні залежності.

При аналізі систем, що проектують, для оцінки живучості необхідна модель, що відображає залежність показників живучості від параметрів системи, що характеризують конфігурацію, мету і умови функціонування, клас розв'язуваних завдань. Адекватність моделі істотно впливає на точність отриманої оцінки.

При дослідженні методів і механізмів забезпечення живучості ОС необхідна оцінка ефективності їх використання, тобто, виникає ряд задач ідентифікації системи (зокрема функціональної ідентифікації), які вирішуються з залученням апарату диференціальних рівнянь, теорії ігор, алгебри, теорії масового обслуговування.

На відміну від завдань аналізу, завдання синтезу в теорії живучості обчислювальних систем формулюється як оптимізаційна задача з відповідною цільовою функцією й обмеженнями. У найбільш загальному вигляді завдання синтезу формулюється наступним чином: визначити при заданій меті функціонування, обмеженнях на параметри системи і клас розв'язуваних завдань конфігурацію ОС (склад її функціональних моделей) S

= (S_1, S_2, …, S_n), яка максимізує відповідні показники живучості, які виступають в якості цільової функції.

У загальній постановці задача синтезу нерозв'язна. Як правило, її спрощують введенням обмежень на клас досліджуваних систем, їх архітектуру, параметри тощо [9, 10]

2. МОДЕЛІ ПОВЕДЕННЯ ВІДМОВОССТІЙКИХ БАГАТОПРОЦЕССОРНИХ СИСТЕМ

2.1 *GL*-моделі ВБС та їх особливості

Одним з методів підвищення надійності ОС є введення відмовостійкості. Питання про побудову та застосування відмовостійких ОС виникає тоді, коли інші шляхи підвищення надійності не можуть забезпечити необхідного рівня надійності з технічних причин або тоді, коли вони виявляються економічно не виправданими.

Складні технічні системи, що працюють в реальному часі, втрата працездатності яких призводить до катастрофічних наслідків, великих економічних втрат, а також системи, що працюють при апріорно не повністю визначених режимах і умовах експлуатації, для ефективного функціонування повинні мати властивості відмовостійкості.

При проектуванні відмовостійких багатопроцесорних систем необхідно вирішувати цілий ряд завдань. До цих завдань, в тому числі, відносяться передбачення надлишкових програмно-апаратних засобів для тестування і виявлення відмови модулів, а потім і для виконання процедур реконфігурації системи, які полягають в інформаційній ізоляції модулів, що відмовили, збереження даних, використанні структурного і тимчасового резервів і відновлення обчислювального процесу.

Для забезпечення надійності ВБС на необхідному рівні, ще на етапі розробки необхідно будувати їх математичні моделі. Побудова таких моделей – це дуже важливе завдання, оскільки вартість таких систем дуже висока, а проведення експериментів над моделями дає можливість оцінити надійнисні характеристики та ефективність ВБС безпосередньо на етапі розробки. Дуже важливо також, щоб вони були адекватні по відношенню до структури і застосовуваним методам підвищення надійності ОС. Тобто, щоб проведення експериментів над побудованою моделлю дозволяло отримати результат заданої точності в прийнятний час, а також дозволяло визначити реакцію системи на появу тієї чи іншої ситуації з відмовою компонента. Без проведення таких експериментів і без ретельного аналізу їх результатів неможливо розробити оптимальну за структурою, ресурсами та якісними показниками систему.

Так, наприклад, недоцільно при розрахунку надійності ВБС великої складності використовувати методи, пов'язані з моделлю двополюсного графа, в яких критерієм працездатності системи є існування шляху між двома полюсами, а аналіз проводиться за умови випадкового існування ребер графа. Це пов'язано з тим, що побудова двополюсного графа аналізованої системи, а також перебір всіх можливих шляхів між його полюсами призводить до значних обчислювальних труднощів, які переводять цю задачу в клас неконструктивних.

В деяких випадках, в якості моделі для розрахунку надійності можна вибрати багатополюсний граф. Такий вибір може бути виправданим, наприклад, для мережевих структур. Критерієм працездатності системи в рамках даної моделі є зв'язність графа, тобто систему вважають справної тільки при наявності зв'язку кожного вузла графа з будь-яким іншим його вузлом. Завдання полягає у визначенні ймовірності того, що хоча б одне дерево ненаправленного графа системи є працездатним. Однак, при аналізі мереж великої складності, комбінаторна задача перебору всіх дерев призводить до значних обчислювальних труднощів, як і у випадку використання двополюсних графів.

Методи статистичного моделювання, що використовують в якості критерію, що характеризує працездатність системи, структурні функції, дозволяють обійти проблему перебору дуже великого числа станів досліджуваної системи за рахунок отримання статистичних оцінок з заданою достовірністю. Але їх недоліком є те, що ці структурні функції мають дуже складний опис. За рахунок цього дані методи значною мірою втрачають свою перевагу.

З наведених вище прикладів можна зробити висновок, що для ефективного аналізу поведінки систем, що мають велике число станів, дуже важливим є компактне подання критерію, згідно з яким здійснюється оцінка поведінки системи в певних умовах і те, що цей критерій залежить від обраної математичної моделі системи.

Як вже говорилося вище, для аналізу поведінки ВБС у потоці відмов у даній роботі використовуються графо-логічні моделі (далі – *GL*-моделі) відмовостійких багатопроцесорних систем. *GL*-моделі вперше були запропоновані в [4]. Основна ідея цього підходу полягає в побудові моделі, що поєднує використання властивостей графів і булевих функцій. Така модель називається графо-логічною. Вона передбачає використання для відображення поведінки системи відповідність між зв'язністю деякого графа і обчислюваними для кожного стану системи значеннями певним чином вибраних булевих функцій. Але, на відміну від відомих моделей, що використовують графи, в даному випадку граф, що використовується, може мати набагато більш просту структуру, ніж структура модельованої системи. Це досягається за рахунок «перенесення» складності на булеві функції, якими позначаються ребра графа. При цьому критерій, використовуваний для визначення стану системи, може бути зроблений досить простим, якщо вдається вибрати регулярну форму представлення булевих функцій в графо-логічній моделі.

Основна перевага даних моделей полягає в тому, що вони дають можливість аналізу поведінки складних ВБС, що включають в себе сотні процесорів у потоці відмов доступними обчислювальними засобами.

Слід нагадати, що під ВБС розуміється відмовостійка багатопроцесорна система, здатна до реконфігурації при появі відмов її модулів – процесорів.

ВБС, стійка до відмов певної кратності, називається базовою. Це означає, що число відмов елементів-процесорів такої системи не повинно перевищувати деяку фіксовану величину, інакше ця система буде нездатною продовжувати функціонування з тими ж параметрами. Базова ВБС, що містить n елементів і є стійкою до відмов, кратність яких не перевищує m $(0 \leq m \leq n)$ позначають як $K(m,n)$.

Необхідно враховувати, що далеко не всі реальні ВБС є базовими. У загальному випадку, ВБС може вийти з ладу при виникненні k певних відмов і, в той же час, бути стійкою до s іншим відмов при $s>k$. Іншими словами, дослідження ВБС не можна обмежувати лише базовими системами, реальний інтерес у багатьох випадках представляє побудова моделей, що відображають поведінку небазових ВБС у потоці відмов. При побудові таких моделей, особливо з урахуванням оптимізації подальшого процесу моделювання, виникає цілий комплекс завдань, пов'язаних з модернізацією як реберних функцій, так і графа базової GL-моделі.

Далі коротко описані основні принципи побудови графо-логічної моделі для базової ВБС $K(m,n)$. Головна ідея полягає у можливості змінювати структуру графа в відповідності з вектором стану елементів системи. Це досягається шляхом приписування ребрам графа визначених булевих функцій.

$K(m,n)$-системі ставиться у відповідність деякий зв'язний неорієнтований граф G, ребра якого позначаються певним чином обраними булевими операторами функціями $f_1(x_1,\ldots, x_n)$, …, $f_L(x_1, \ldots, x_n)$, де кожна змінна x_i $(i = \overline{1,n})$ є індикаторною функцією, що позначає стан i-го елемента системи. Тобто $x_i = 1$, якщо і-й елемент системи працездатний, і $x_i = 0$ в іншому випадку. Ребро, позначене функцією $f_j(x_1, \ldots, x_n)$, залишається у графі G, якщо $f_j(x_1, \ldots, x_n) = 1$ і видаляється з нього, якщо $f_j(x_1, \ldots, x_n) = 0$.

Якщо вибрати функції f1$(x_1,\ldots, x_n)$, …, $f_L(x_1, \ldots, x_n)$ таким чином, щоб граф G втрачав зв'язність тільки при рівності нулю не менше $m+1$ будь-яких з n змінних, то відмова n-процесорної системи можна інтерпретувати як втрату зв'язності графа G. Іншими словами, функції $f_1,\ldots, f_L$ повинні мати наступну властивість: для всіх n-розрядних наборів, що містять будь-яке поєднання з не менше, ніж $m+1$ нульових розрядів, знайдеться r $(r < L)$

функцій з множини ($f_1,\ldots, f_L$), які приймають нульові значення на цих наборах і позначають r таких ребер, видалення яких з графа G призводить до втрати його зв'язності.

Очевидно, що дуже важливими етапами при побудові графо-логічної моделі є вибір графа G і функцій $f_1,\ldots, f_L$.

Найпростішим графом є незамкнутий ланцюг, а найпростішими функціями – $f_i = x_i$.. Такий граф з ребрами, позначеними функціями $f_i = x_i$., може бути використаний в графо-логічної моделі для $K(0,n)$ – системи. Якщо замкнути ланцюг, тобто отримати циклічний граф, то разом з такими ж функціями $f_i = x_i$ отриманий граф стане графо-логічною моделлю $K(1,n)$-системи. На такі циклічні графи звертається увага в ряді робіт (наприклад, [14]).

Далі розглянемо окремо принципи побудови GL-моделей для базових $K(m, n)$-систем для випадків, коли $m = 2$ і коли $m > 2$.

2.2 Формування *GL*-моделей для базових *K*(2, *n*) систем

Розглянемо випадок, коли система містить n модулів і стає непрацездатною тільки тоді, коли відмовляють не менше трьох процесорів. За прийнятою термінологією така система є $K(2,n)$ – системою. В якості графа G обраний неорієнтований циклічний n-реберний граф (рис. 1.1), i-му ребру якого відповідає булева функція

$$f_i\,(x_1,\,\ldots,\,x_n) = \prod_{j=1,\left[\frac{n-1}{2}\right]} (x_i \vee x_{(i+j)\,\mathrm{mod}(n)}), \qquad (2.1)$$

для $i = (\overline{1,n})$.

Нехай також i-е ребро існує, якщо f_i=1, і відсутнє, якщо f_i =0.

Граф G втрачає зв'язність, якщо і тільки якщо не менше трьох змінних з множини {x1,..., xn} приймають нульові значення.

Покажемо це.

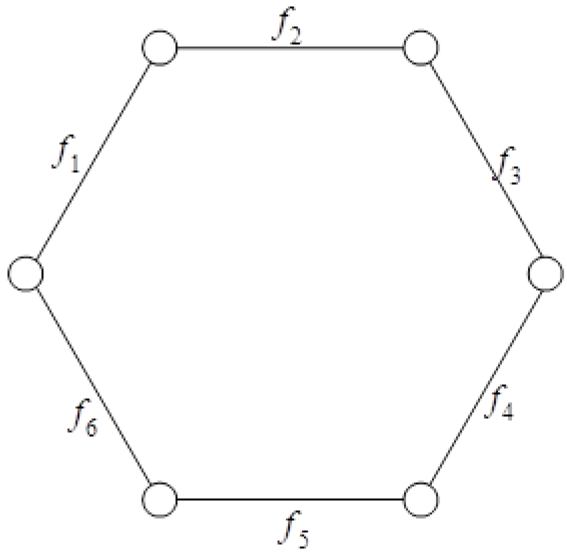


Рисунок 1.3 – Зв’язність графа G

Спочатку необхідно показати, що при рівності нулю менше трьох змінних, нульове значення може прийняти тільки одна функція.

Очевидно, що при рівності нулю однієї змінної $f_i \neq 0$ для будь-якого $i = (\overline{1,n})$

Нехай $x_p, x_q \in \{x_1, ..., x_n\}$ і тільки вони рівні нулю. Необхідно відмітити, як очевидне, той факт, що в цьому випадку із усіх f_i, $i = (\overline{1,n})$, тільки f_p і f_g можуть бути рівні нулю. Очевидно також, що $f_p = 0$, якщо виконується

$$q \in \left\{ (p+1)\operatorname{mod}(n), (p+2)\operatorname{mod}(n), ..., \left(p + \left[\frac{n-1}{2} \right] \right) \operatorname{mod}(n) \right\}, \tag{1}$$

а $f_q = 0$, якщо виконується

$$p \in \left\{ (q+1)\operatorname{mod}(n), (q+2)\operatorname{mod}(n), ..., \left(q + \left[\frac{n-1}{2} \right] \right) \operatorname{mod}(n) \right\} \tag{2}$$

Легко показати, що одночасне виконання (1) і (2) при $p \neq q$ неможливо. Отже, висловлене припущення справедливо.

Тепер необхідно показати, що для трьох і більше рівних нулю змінних з $\{x_1,...,x_n\}$ принаймні дві функції f_k и f_l $(k = \overline{1,n},\ l = \overline{1,n},\ k \neq l)$ приймають нульові значення.

Нехай x_p, x_q, $x_r \in \{x_1,...,x_n\}$ і $p < q < r$. Якщо виконується (1), то $f_p = 0$. При цьому виповняється або

$$r \in \left\{ (q+1)\,\mathrm{mod}(n), (q+2)\,\mathrm{mod}(n), ..., \left(q + \left[\frac{n-1}{2} \right] \right) \mathrm{mod}(n) \right\}, \tag{3}$$

і тоді $f_q=0$, або (3) не виконується, і тоді, враховуючи, що $r + \left[\frac{n-1}{2} \right] > n$, легко помітити, що

$$p \in \left\{ (r+1)\,\mathrm{mod}(n), (r+2)\,\mathrm{mod}(n), ..., \left(r + \left[\frac{n-1}{2} \right] \right) \mathrm{mod}(n) \right\}. \tag{4}$$

Отже, $f_r=0$.

Якщо умова (2) не виконується(при цьому очевидно, що $r \notin \left\{ (p+1)\,\mathrm{mod}(n), ..., \left(p + \left[\frac{n-1}{2} \right] \right) \mathrm{mod}(n) \right\}$), то $f_p \neq 0$. Але в цьому випадку очевидне виконання умов (3) та (4). Отже, $f_q = 0$ і $f_r = 0$, що ми і хотіли показати.

Наведене дозволяє легко будувати графо-логічні моделі для будь-яких значень n при $m=2$. Наприклад, для випадку $n = 6$, вона має вигляд, представлений на рисунку 2.1, де функції

$$\begin{aligned} f_1 &= x_1 \vee x_2 \cdot x_3 \\ f_2 &= x_2 \vee x_3 \cdot x_4 \\ f_3 &= x_3 \vee x_4 \cdot x_5 \\ f_4 &= x_4 \vee x_5 \cdot x_6 \\ f_5 &= x_5 \vee x_6 \cdot x_1 \\ f_6 &= x_6 \vee x_1 \cdot x_2 \end{aligned}$$

Видно, что функція f_{i+1} виходить з функції f_i шляхом додавання 1 по mod n до індексів змінних. Число ребер в точності дорівнює n.

Простота формування моделі робить її дуже привабливою. Але цей спосіб формування не можна поширити для випадків $m > 2$, тобто пряме узагальнення не має місця і для побудови моделей таких систем потрібно керуватися іншими принципами. Вони і будуть розглянуті нижче

2.3 Загальний метод формування *GL*-моделей для *K*(*m, n*)-систем

Існує більш загальний підхід до розв'язання задачі побудови графологічної моделі ВБС, стійкої до відмов не менше m модулів, де $m > 2$. Розглянемо його.

В попередньому параграфі відзначалися, з одного боку, необхідність відшукання загального рішення для довільних значень числа модулів і числа захищених відмов і, з іншого боку, неможливість прямого узагальнення отриманих там результатів. Тому в цьому параграфі розглянемо рішення поставленої задачі за допомогою іншого підходу.

Нехай є k -ВБС (тобто ВБС, захищена від відмов кратності не вище k), що складається з n елементів (модулів). Стан системи (відмовна ситуація) визначається поєднанням m відмовивших і n-m справних елементів. Кожному стану системи можна поставити у відповідність деяку неупорядковану m-вибірку з n елементів. Надалі будемо говорити просто m-вибірка, маючи на увазі, що це невпорядкована вибірка.

Нехай $n_u = \{\alpha_1, \alpha_2, ..., \alpha_{n_u}\}$ – деяка підмножина множини всіх елементів k-ВБС. Для k-ВБС визначимо множину $K(m_v, n_u)$ як множина, що складається з усіх m_v вибірок з n_u елементів.

Нехай $K(s, n_p) = \{a_1, a_2, ..., a_g\}$, де a_i – деяка S-вибірка з n_p елементів ($i = \overline{1, g}$), а $K(t, n_r) = \{b_1, b_2, ..., b_h\}$, де b_j – деяка t-вибірка з n_r елементів. Добуток множин $K(s, n_p)$ і $K(t, n_r)$ визначимо у відповідності з наступним чином:

$K(s, n_p) \times K(t, n_r) = \{(a_1 \times b_1), (a_1 \times b_2), \dots, (a_1 \times b_h), \dots, (a_g \times b_1), (a_g \times b_2), \dots, (a_g \times b_h)\}$, де $(a_i \times b_j)$ - результат конкатенації елементів вибірок a_i і b_j.

Далі розіб'ємо n-множину, що представляє мнодину всіх елементів системи, на q непересічних підмножин таким чином, що в 1-ій підмножині буде n_1 елементів, у 2-му – n_2 елементів і т. д., у q-му – n_q елементів, тобто $n = \sum_{i=1}^{q} n_i$. Тоді, за аналогією з відомою комбінаторною тотожністю,

$$\sum_{m_1,\ldots,m_q} C_{n1}^{m1} \cdot C_{n2}^{m2} \cdot C_{nq}^{mq} = \cdot C_n^m \qquad (2.2)$$

можна записати наступну рівність

$$K(m,n) = \bigcup_{m_1,m_2,\ldots,m_q} K(m_1,n_1) \times K(m_2,n_2) \times \ldots \times K(m_q,n_q) \qquad (2.3)$$

де операція сумми в виразі(2.2) і операція об'єднання множин у виразі (2.3) виконуються для всіх таких наборів чисел $m_1, m_2, \ldots, m_q$, для яких має місце $m = \sum_{i=1}^{q} m_i$ и $0 \le m_i \le n_i$.

Далі, використовуючи розбиття множини, представлене співвідношенням (2.3), займемося побудовою *GL*-моделі *k*-ВБС на основі графа кільцевого типу для довільних значень *k*. Враховуючи, що граф кільцевого типу втрачає зв'язність тільки тоді, коли видаляються не менше двох ребер, легко встановлюється критерій відмови *k*-ВБС в термінах множини булевих функцій, якими позначені ребра цього графа. Таким критерієм є рівність нулю значень не менш ніж двох функцій. Для того, щоб визначити процедуру вибору таких функцій, попередньо покажемо наступне.

Нехай *m*-ВБС має *n* елементів, і множина, складена з цих елементів, розбита на *q* підмножин з числом елементів $n_1, n_2, \ldots, n_q$ відповідно. Нехай також існують $f_1(x_1,\ldots, x_n), \ldots, f_L(x_1, \ldots, x_n)$ – такі булеві функції, що кожна з функцій f_j $(j \in \overline{1,L})$ приймає нульове значення, якщо дорівнюють нулю, принаймні, *m* змінних, що визначають стан m_i елементів для кожної з n_i – підмножин $(i \in \overline{1,q})$ у відповідності зі значеннями чисел $m_1, m_2, \ldots, m_q$ з *i*-го набору. Тоді справедливо наступне: не менше двох функцій з мнодини $f_1, \ldots, f_L$ приймуть нульові значення, якщо, принаймні, будь-які $m+1$ з *n* змінних дорівнюють нулю. Покажемо це.

Спочатку покажемо, що при рівності нулю менше, ніж $m+1$ змінних, нульове значення може прийняти тільки одна функція $f_1, \ldots, f_L$. Для цього достатньо розглянути випадок, коли рівно *m* змінних дорівнюють нулю.

Враховуючи, що будь-яка m-вибірка з n змінних відповідає тільки одному набору значень m_1, m_2, …, m_q, висловлене припущення стає очевидним.

Перейдемо тепер до випадку, коли $m+1$ змінних приймають нульове значення. Нехай це будуть наступні змінні:

$$x_{\alpha_1} = 0, x_{\alpha_2} = 0, ..., x_{\alpha_m} = 0, x_{\alpha_{m+1}} = 0,$$

де числа (індекси α_1, α_2,…, α_m, α_{m+1}) представляють будь-яку (m+1) – вибірку індексів з n можливих.

З елементів цієї (m+1) – вибірки можливо скласти $C_{m+1}^{m} = \dfrac{(m+1)!}{m!(m+1-m)!} = m+1$ різних m – вибірок. Розглянемо наступні три із них: (α_1, α_2,…, α_{m-1}, α_m), (α_1, α_2,…, α_{m-1}, α_{m+1}) і (α_2, …, α_{m-1}, α_m, α_{m+1}).

Далі для простоти будемо розуміти під вибіркою індексів (α_1, α_2,…) вибірку змінних ($x_{\alpha_1}, x_{\alpha_2}, ...$) і, відповідно, під елементом α_i-елемент x_{α_i}.

Нехай вибірка (α_1, α_2, …, α_m) відповідає деякому набору (причому, цей набір єдиний) значень m_1, m_2, …, m_q з номером j. Покажемо, що хоча б одна з вибірок α_1, α_2,…, α_{m-1}, α_{m+1} и α_2, …, α_{m-1}, α_m, α_{m+1} не може відповідати набору з тим же номером j, який був встановлений для вибірки α_1, α_2, …, α_m. Дійсно, якщо з структури вибірки α_1, α_2,…, α_{m-1}, α_{m+1} випливає, що α_{m+1} не потрапляє в підмножину n_i, що містить елемент α_m, то, очевидно, відбувається зміна набору m_1, m_2, …, m_q. У будь-якому випадку, знаходиться α_{m+1} в одному домені разом з одним з елементів α_1, α_2,…, α_{m-1}, або ні – відбудеться збільшення на одиницю значення деякого розряду набору m_1, m_2, …, m_q. Якщо ж виявляється, що α_{m+1} знаходиться в одній підмножині n_i разом з елементом α_m, то вибірці (, α_2,…, α_{m-1}, α_{m+1}) відповідає той же набір з номером j, що і вибірці (α_1, α_2, …, α_m). Але при цьому третя вибірка (α_2, …, α_{m-1}, α_m, α_{m+1}) дає зміну набору, так як значення розряду m_i набору m_1, m_2, …,m_q збільшується на одиницю.

Таким чином, три розглянутих вибірки повинні відповідати не менш, ніж двом різним наборам значень чисел m_1, m_2, …,m_q, а це означає, що серед функцій f_1, …, f_L знайдуться, принаймні, дві функції, які візьмуть нульові значення. Що і потрібно було показати.

Покажемо, що будь-яка функція f_i з розглянутої множини функцій $(x_1,\dots, x_n)$, …, $f_L(x_1, \dots, x_n)$, що задовольняють умовам теореми, може бути представлена в наступному вигляді:

$$f_i = f_{m_1} \vee f_{m_2} \vee \dots \vee f_{m_q} \ , \tag{2.4}$$

де

$$f_{m_j} = y_{a_1} \wedge y_{a_2} \wedge \dots \wedge y_{a_{N\,j}}; \ \ j = \overline{1,q};\ N_j = C_{n_j}^{m_j};$$

$$y_{a_t} = x_{\alpha_1} \vee x_{\alpha_2} \vee \dots\ x_{\alpha_{mj}};\ \ t = \overline{1,\ C_{n_j}^{m_j}}\ \ .$$

Зазначимо також, що функції f_{m_j} - це функції, що приймають нульові значення тільки тоді, коли дорівнюють нулю всі змінні хоча б з однієї m_j - вибірки з n_j - підмножини.

Дійсно, нехай $(x_{\alpha 1}, x_{\alpha 2},\dots, x_{\alpha_m})$ - деякий набір m змінних з n – множини, і кожна змінна в цьому наборі приймає нульові значення. Очевидно, що в цьому наборі є група змінних, що представляє яку-небудь m_1-вибірку з n_1 – підмножини, група змінних, що представляє яку-небудь m_2-вибірку з n_2 – підмножини, і так далі до групи змінних, що представляють яку-небудь m_q - вибірку з n_q – підмножини. Групи змінних, що становлять цей набір, відповідають деякому єдиному набору чисел $(m_1,\ m_2,\ \dots,\ m_q)$. Нехай, наприклад, m_1-вибірка представляє набір змінних $(x_{\alpha 1},\ x_{\alpha 2},\dots,\ x_{\alpha_{m1}})$. Тоді справедливо наступне:

$$y_{a_1} = x_{\alpha_1} \vee x_{\alpha_2} \vee \dots\ x_{\alpha_{m1}} = 0,$$

$$f_{m_1} = y_{a_1} \wedge y_{a_2} \wedge \dots \wedge y_{a_{N_1}} = 0,\ (N_1 = C_{n_1}^{m_1})\,.$$

Аналогічно, розглядаючи набори змінних, які складають m_2- вибірку і т. д. m_q - вибірку, отримаємо, що $f_{m2}=0,\dots f_{mq}=0$. Отже, у цьому випадку функція f_i з виразу (2.4) приймає нульове значення.

Розглянемо тепер набір змінних, групи змінних якого відповідають деякого набору чисел $\boldsymbol{m}' = (m_1', m_2', \dots, m_q')$, відмінному від розглянутого вище набору $\boldsymbol{m} = (m_1, m_2, ..., m_q)$. Якщо набори $\boldsymbol{m}$ і $\boldsymbol{m}^/$ відрізняються, то для того, щоб виконувались рівності

$$\sum_{i=1}^{q} m_i = m,$$

$$\sum_{i=1}^{q} m_i^/ = m,$$

необхідно, щоб виконувалися також дві наступні нерівності: $m_p^/ < m_p$ і $m_r^/ > m_r$, де p и r $(p \neq r)$ - деякі цілі числа, що задовольняють нерівності $1 \leq p \leq q$ $і$ $1 \leq r \leq q$. Звідси випливає, що група змінних, складова будь- $m_r^/$ - вибірку з n_r – множини містить хоча б одну змінну, значення якої дорівнює одиниці. При цьому отримаємо:

$$y_{a_1} = 1,\ y_{a_2} = 1, ..., y_{a_{N_j}} = 1;\ f_{m_j} = 1;\ f_i = 1.$$

Нарешті, якщо число змінних у цьому наборі менше числа m, то це означає, що в одній з підмножин, наприклад в $n_з$ – підмножині будуть змінні, що мають одиничні значення. Відповідно отримаємо f_i = 1.

Таким чином, функція f_i приймає нульове значення тільки в тому випадку, якщо не менш m змінних приймають нульові значення. Причому, ці змінні повинні представляти m_1- вибірки, m_2-вибірки,..., m_q –вибірки з n_1 – підмножини, n_2 – підмножини,..., n_q – підмножини відповідно, для єдиного набору чисел $(m_1, m_2, ..., m_q)$. Висновок: функція f_i задовольняє умовам, описаним вище.

При побудові *GL*-моделі, що описує поведінку *k*-ВБС, запис булевих функцій $f_1(x_1,\dots, x_n), \dots, f_L(x_1, \dots, x_n)$, що відповідають описаним вище умовам, зручно виконати за наступним правилом:

1. Провести розбиття множини з n – елементів, складових *k*-ВБС, на q непересічних підмножин з числом елементів, відповідно $n_1, n_2, ..., n_q$ $(\sum_{i=1}^{q} n_i = n)$.
2. Перерахувати всі можливі набори значень $m_1, m_2, ..., m_q$ за умови, що

$$\sum_{i=1}^{q} m_i = m, m_i \le n_i.$$

3. Для кожного з наборів значень чисел m1 $m_1, m_2, ..., m_q$ записати вираз виду

$$K(m_1, n_1) \times K(m_2, n_2) \times ... \times K(m_q, n_q) =$$
$$= \{(a_1 \times b_1 \times ... \times c_1), (a_1 \times b_1 \times ... \times c_2), ..., (a_{n_1} \times b_{n_2} \times ... \times c_{n_q})\},$$

де a_i, b_j ,…, c_k – це m_1 – вибірки з n_1 – множини, m_2 – вибірки з n_2 – множини, …, m_q – вибірки з n_q – множини ($i = \overline{1, C_{n_1}^{m_1}}$, $j = \overline{1, C_{n_2}^{m_2}}$, ..., $k = \overline{1, C_{n_q}^{m_q}}$), тобто a_i вибірка виду $\{\alpha_1, \alpha_2, …, \alpha_{m1}\}$, b_j вибірка виду $\{\beta_1, \beta_2, …, \beta_{m2}\}$, …, c_k вибірка виду $\{\gamma_1, \gamma_2, …, \gamma_{mq}\}$.

4. Здійснити перехід від виразу з п. 3 до булевих виразів. Булева функція $f_i(x_1,…, x_n)$, де i – номер набору значень чисел m_1, $m_2, ..., m_q$ $(i = \overline{1, L})$, записується у вигляді:

$$f_i = (\bigwedge_i y_{a_i}) \bigvee (\bigwedge_j y_{b_j}) \bigvee \dots \bigvee (\bigwedge_k y_{c_k})$$

где $y_{a_i} = \bigvee_{\sigma=1}^{m_1} x_{\alpha_\sigma}$, $y_{\beta_j} = \bigvee_{\theta=1}^{m_2} x_{\beta_\theta}$, ..., $y_{c_k} = \bigvee_{\omega=1}^{m_q} x_{\gamma_\omega}$.

Розглянемо варіант розбиття n – множини, при якому в певній мірі спрощується запис функцій f_1, …, f_L. Такий варіант розбиття буде досягнуто, коли у виразі (2.2) для всіх підмножин $K(m_j, n_j)$ має місце наступне: або m_j = 1, або $m_j = n_j$. Якщо $m_j = 1$, отримаємо вираз (2.4) $y_{a_t} = x_{a_t}, t = \overline{1, n_j}$ і, отже, вирази для f_{mj} становлять елементарні кон'юнкції $f_{m_j} = x_{\alpha_1} \wedge x_{\alpha_2} \wedge ... \wedge x_{\alpha_{n_j}}$. Якщо $m_j = n_j$, отримаємо вираз (2.4) $y_{a_t} = x_{a_t}, t = \overline{1, n_j}$ і, отже, вирази для f_{mj} становлять елементарні диз'юнкції $f_{m_j} = x_{\alpha_1} \vee x_{\alpha_2} \vee ... \vee x_{\alpha_{nj}}$. Відповідно, при виборі розглянутого варіанта розбиття n – множини отримуємо функції f_i у виразах (2.4) з більш простою формою запису. Тому розбиття n – множини доцільно здійснювати у декілька етапів. На першому етапі отримуємо n_1, $n_2, ..., n_q$ підмножини. Після першого етапу розбиття подальше розбиття здійснюємо тільки для таких підмножин n_i, вираження яких не відповідає $K(1, n_j)$ або $K(n_j, n_j)$.

Приклад

Нехай дана система являє собою 3-ВБС і складається з 8 елементів, позначених номерами 1, 2 , 3, 4, 5, 6, 7, 8. Необхідно побудувати *GL* - модель для даної системи, використовуючи граф кільцевого типу.

Розіб'ємо множину елементів системи на 3 підмножини:

$n_1 = \{1, 2, \}$, $n_2 = \{3, 4, 5\}$ і $n_3 = \{6, 7, 8\}$. У відповідності з виразом (2.2), отримаємо для множини $K(3, 8)$:

$$K(m,n) = \bigcup_{m_1,m_2,m_3} K(m_1,n_1) \times K(m_2,n_2) \times K(m_3,n_3),$$

де операція об'єднання проводиться для всіх наборів m_1, m_2, m_3, причому, $\sum_{i=1}^{3} m_i = 3$, $0 \le m_i \le 3$. Для визначення множин $K(m_i, n_i)$ складемо таблицю усіх можливих наборів чисел m_1, m_2, m_3 для умов даного прикладу.

Таблиця 2.1

№ набору	Значення чисел		
	m_1	m_2	m_3
1	0	0	3
2	0	1	2
3	0	2	1
4	0	3	0
5	1	0	2
6	1	1	1
7	1	2	0
8	2	0	1
9	2	1	0

Тепер віраз для $K(3, 8)$ можна записати наступним чином:

$K(3, 8) = (K(0, n_1) \times K(0, n_2) \times K(0, n_3)) \cup (K(0, n_1) \times K(1, n_2) \times K(2, n_3)) \cup$

$\cup (K(0, n_1) \times K(2, n_2) \times K(1, n_3)) \cup \ldots \cup (K(2, n_1) \times K(1, n_2) \times K(0, n_3))$.

В цей вираз входять пусті множини $K(0, n_i)$ для всіх $i = \overline{1,3}$, а також множини

$K(3, n_3) = \{(678)\}$, $K(1, n_2) = \{(3), (4), (5)\}$, $K(2, n_3) = \{(67), (78), (68)\}$,

$K(2, n_2) = \{(34), (45), (35)\}$, $K(1, n_3) = \{(6), (7), (8)\}$, …, $K(2, n_1) = \{(12)\}$.

З виразів для множин $K(m_j, n_j)$ та з таблиці 2.1 видно, що не всі ті множини, які вийшли після розбиття n-множини, являють собою множини типу $K(n_j, n_j)$ або $K(1, n_j)$. Так, для наборів чисел m_1, m_2, m_3 з номерами 2, 3, 5 і 7, вийшли множества виду $K(2, 3)$. Отже, потрібно провести подальше розбиття цих множин. Його результат представлений в таблиці 2.2.

Таблиця 2.2

<table>
<tr><td rowspan="4">№ наборов</td><td colspan="2">$n_1 = \{1, 2\}$</td><td colspan="2">$n_2 = \{3, 4, 5\}$</td><td colspan="2">$n_3 = \{6, 7, 8\}$</td></tr>
<tr><td>n_{1_1}</td><td>n_{1_2}</td><td>n_{2_1}</td><td>n_{2_2}</td><td>n_{3_1}</td><td>n_{3_2}</td></tr>
<tr><td colspan="2">m_1</td><td colspan="2">m_2</td><td colspan="2">m_3</td></tr>
<tr><td>m_{1_1}</td><td>m_{1_2}</td><td>m_{2_1}</td><td>m_{2_2}</td><td>m_{3_1}</td><td>m_{3_2}</td></tr>
<tr><td>1</td><td colspan="2">0</td><td colspan="2">0</td><td colspan="2">3</td></tr>
<tr><td>2</td><td colspan="2">0</td><td colspan="2">1</td><td>1</td><td>1</td></tr>
<tr><td>3</td><td colspan="2">0</td><td colspan="2">1</td><td>2</td><td>0</td></tr>
<tr><td>4</td><td colspan="2">0</td><td colspan="2">0</td><td colspan="2">3</td></tr>
<tr><td>5</td><td colspan="2">1</td><td colspan="2">0</td><td>1</td><td>1</td></tr>
<tr><td>6</td><td colspan="2">1</td><td colspan="2">0</td><td>2</td><td>0</td></tr>
<tr><td>7</td><td colspan="2">1</td><td colspan="2">1</td><td colspan="2">1</td></tr>
<tr><td>8</td><td colspan="2">1</td><td>1</td><td>1</td><td colspan="2">0</td></tr>
</table>

<table>
<tr><td rowspan="4">№ наборов</td><td colspan="2">$n_1 = \{1, 2\}$</td><td colspan="2">$n_2 = \{3, 4, 5\}$</td><td colspan="2">$n_3 = \{6, 7, 8\}$</td></tr>
<tr><td>n_{1_1}</td><td>n_{1_2}</td><td>n_{2_1}</td><td>n_{2_2}</td><td>n_{3_1}</td><td>n_{3_2}</td></tr>
<tr><td colspan="2">m_1</td><td colspan="2">m_2</td><td colspan="2">m_3</td></tr>
<tr><td>m_{1_1}</td><td>m_{1_2}</td><td>m_{2_1}</td><td>m_{2_2}</td><td>m_{3_1}</td><td>m_{3_2}</td></tr>
<tr><td>9</td><td colspan="2">1</td><td>2</td><td>0</td><td colspan="2">0</td></tr>
<tr><td>10</td><td colspan="2">2</td><td colspan="2">0</td><td colspan="2">1</td></tr>
<tr><td>11</td><td colspan="2">2</td><td colspan="2">1</td><td colspan="2">0</td></tr>
</table>

Перейдемо тепер до отримання булевих функцій для *GL*-моделі. Використовуючи правило запису булевих функцій для *GL*-моделі, наведене вище, для кожного набору чисел m_1, m_2, m_3 і для кожного набору m_{1_1}, m_{1_2}, m_{2_1}, m_{2_2}, m_{3_1}, m_{3_2} з таблиці 2.2 отримуємо наступні булеві функції:

$$f_1 = x_6 \vee x_7 \vee x_8$$

$$f_2 = x_3x_4x_5 \vee x_6x_7 \vee x_8$$

$$f_3 = x_3x_4x_5 \vee x_6 \vee x_7$$

$$f_4 = x_6 \vee x_7 \vee x_8$$

$$f_5 = x_1x_2 \vee x_6x_7 \vee x_8$$

$$f_6 = x_1x_2 \vee x_6 \vee x_7$$

$$f_7 = x_1x_2 \vee x_3x_4x_5 \vee x_6x_7x_8$$

$$f_8 = x_1x_2 \vee x_3x_4 \vee x_5$$

$$f_9 = x_1x_2 \vee x_3 \vee x_4$$

$$f_{10} = x_1 \vee x_2 \vee x_6x_7x_8$$

$$f_{11} = x_1 \vee x_2 \vee x_3x_4x_5$$

Розглянута методика вирішує задачу побудови *GL*-моделі для *k*-ВБС при довільних значеннях *k*. Її особливістю є регулярний спосіб побудови булевих функцій, які використовуються для позначення ребер кільцевого графа, на основі якого будується дана модель.

Наведені правила і наступна за ними методика можуть здатися занадто складними і малопридатними для практичного використання. Але, насправді, це не так, оскільки знайдені вирази для $K(m, n)$ можна використовувати при отриманні виразів $K(s, r)$, де $s \geq m$, а $r \geq n$. Продемонструємо це на прикладі побудови GL-моделі для $K(5, 10)$ двома способами.

1-*ий спосіб*

Як було зазначено вище, найбільш простим буде формування булевих виразів для $K(1, i)$ або $K(i, i)$. Використовуючи цю властивість, можна відразу записати реберні функції, розбивши множину змінних

$\{x_1, x_2, \ldots, x_{10}\}$ на 5 підмножин $\{x_1, x_2\}$, $\{x_3, x_4\}$, $\{x_5, x_6\}$, $\{x_7, x_8\}$ і $\{x_9, x_{10}\}$. Для кожної з цих підмножин можемо записати

$$K(1, 2) = x_i\, x_{i+1} \tag{2.5}$$

$$K(2, 2) = x_i \vee x_{i+1}. \tag{2.6}$$

Тепер перераховуємо сукупності $K(i, j)$, які відповідають всім можливим варіантам розподілу відмов по підмножинам (див. табл. 2.3), і відповідні функції.

Таблиця 2.3

$\{x_1, x_2\}$	$\{x_3, x_4\}$	$\{x_5, x_6\}$	$\{x_7, x_8\}$	$\{x_9, x_{10}\}$	
$K(1,2)$	$K(1,2)$	$K(1,2)$	$K(1,2)$	$K(1,2)$	$f_1 = x_1x_2 \vee x_3x_4 \vee x_5x_6 \vee x_7x_8 \vee x_9x_{10}$
$K(2,2)$		$K(1,2)$	$K(1,2)$	$K(1,2)$	$f_2 = x_1 \vee x_2 \vee x_5x_6 \vee x_7x_8 \vee x_9x_{10}$
$K(2,2)$	$K(1,2)$	$K(1,2)$	$K(1,2)$		$f_3 = x_1 \vee x_2 \vee x_5x_6 \vee x_7x_8 \vee x_9x_{10}$
$K(2,2)$	$K(1,2)$		$K(1,2)$	$K(1,2)$	$f_4 = x_1 \vee x_2 \vee x_3x_4 \vee x_7x_8 \vee x_9x_{10}$
$K(1,2)$	$K(2,2)$		$K(1,2)$	$K(1,2)$	$f_5 = x_1x_2 \vee x_3 \vee x_4 \vee x_7x_8 \vee x_9x_{10}$
$K(1,2)$	$K(2,2)$	$K(1,2)$		$K(1,2)$	$f_6 = x_1x_2 \vee x_3 \vee x_4 \vee x_5x_6 \vee x_9x_{10}$
$K(1,2)$	$K(2,2)$	$K(1,2)$	$K(1,2)$		$f_7 = x_1x_2 \vee x_3 \vee x_4 \vee x_5x_6 \vee x_7x_8$
$K(1,2)$		$K(2,2)$	$K(1,2)$	$K(1,2)$	$f_8 = x_1x_2 \vee x_5 \vee x_6 \vee x_7x_8 \vee x_9x_{10}$
$K(1,2)$	$K(1,2)$	$K(2,2)$		$K(1,2)$	$f_9 = x_1x_2 \vee x_3x_4 \vee x_5 \vee x_6 \vee x_9x_{10}$
$K(1,2)$	$K(1,2)$	$K(2,2)$	$K(1,2)$		$f_{10} = x_1x_2 \vee x_3x_4 \vee x_5 \vee x_6 \vee x_7x_8$
$K(1,2)$		$K(1,2)$	$K(2,2)$	$K(1,2)$	$f_{11} = x_1x_2 \vee x_5x_6 \vee x_7 \vee x_8 \vee x_9x_{10}$

$\{x_1, x_2\}$	$\{x_3, x_4\}$	$\{x_5, x_6\}$	$\{x_7, x_8\}$	$\{x_9, x_{10}\}$	
$K(1,2)$	$K(1,2)$		$K(2,2)$	$K(1,2)$	$f_{12} = x_1x_2 \vee x_3x_4 \vee x_7 \vee x_8 \vee x_9x_{10}$
$K(1,2)$	$K(1,2)$	$K(1,2)$	$K(2,2)$		$f_{13} = x_1x_2 \vee x_3x_4 \vee x_5x_6 \vee x_7 \vee x_8$
$K(1,2)$		$K(1,2)$	$K(1,2)$	$K(2,2)$	$f_{14} = x_1x_2 \vee x_5x_6 \vee x_7x_8 \vee x_9 \vee x_{10}$
$K(1,2)$	$K(1,2)$		$K(1,2)$	$K(2,2)$	$f_{15} = x_1x_2 \vee x_3x_4 \vee x_7x_8 \vee x_9 \vee x_{10}$
$K(1,2)$	$K(1,2)$	$K(1,2)$		$K(2,2)$	$f_{16} = x_1x_2 \vee x_3x_4 \vee x_5x_6 \vee x_9 \vee x_{10}$
	$K(2,2)$	$K(1,2)$	$K(1,2)$	$K(1,2)$	$f_{17} = x_3 \vee x_4 \vee x_5x_6 \vee x_7x_8 \vee x_9x_{10}$
	$K(1,2)$	$K(2,2)$	$K(1,2)$	$K(1,2)$	$f_{18} = x_3x_4 \vee x_5 \vee x_6 \vee x_7x_8 \vee x_9x_{10}$
	$K(1,2)$	$K(1,2)$	$K(2,2)$	$K(1,2)$	$f_{19} = x_3x_4 \vee x_5x_6 \vee x_7 \vee x_8 \vee x_9x_{10}$
	$K(1,2)$	$K(1,2)$	$K(1,2)$	$K(2,2)$	$f_{20} = x_3x_4 \vee x_5x_6 \vee x_7x_8 \vee x_9 \vee x_{10}$
$K(2,2)$	$K(2,2)$	$K(1,2)$			$f_{21} = x_1 \vee x_2 \vee x_3 \vee x_4 \vee x_5x_6$
$K(2,2)$	$K(2,2)$		$K(1,2)$		$f_{22} = x_1 \vee x_2 \vee x_3 \vee x_4 \vee x_7x_8$
$K(2,2)$	$K(2,2)$			$K(1,2)$	$f_{23} = x_1 \vee x_2 \vee x_3 \vee x_4 \vee x_9x_{10}$
$K(2,2)$		$K(2,2)$		$K(1,2)$	$f_{24} = x_1 \vee x_2 \vee x_5 \vee x_6 \vee x_9x_{10}$
$K(2,2)$		$K(2,2)$	$K(1,2)$		$f_{25} = x_1 \vee x_2 \vee x_5 \vee x_6 \vee x_7x_8$
$K(2,2)$	$K(1,2)$	$K(2,2)$			$f_{26} = x_1 \vee x_2 \vee x_5 \vee x_6 \vee x_3x_4$
$K(2,2)$			$K(2,2)$	$K(1,2)$	$f_{27} = x_1 \vee x_2 \vee x_7 \vee x_8 \vee x_9x_{10}$
$K(2,2)$		$K(1,2)$	$K(2,2)$		$f_{28} = x_1 \vee x_2 \vee x_5x_6 \vee x_7 \vee x_8$
$K(2,2)$	$K(1,2)$		$K(2,2)$		$f_{29} = x_1 \vee x_2 \vee x_3x_4 \vee x_7 \vee x_8$
$K(2,2)$			$K(1,2)$	$K(2,2)$	$f_{30} = x_1 \vee x_2 \vee x_7x_8 \vee x_9 \vee x_{10}$
$K(2,2)$		$K(1,2)$		$K(2,2)$	$f_{31} = x_1 \vee x_2 \vee x_5x_6 \vee x_9 \vee x_{10}$
$K(2,2)$	$K(1,2)$			$K(2,2)$	$f_{32} = x_1 \vee x_2 \vee x_3x_4 \vee x_9 \vee x_{10}$
	$K(2,2)$	$K(2,2)$		$K(1,2)$	$f_{33} = x_3 \vee x_4 \vee x_5 \vee x_6 \vee x_9x_{10}$
	$K(2,2)$	$K(2,2)$	$K(1,2)$		$f_{34} = x_3 \vee x_4 \vee x_5 \vee x_6 \vee x_7x_8$
$K(1,2)$	$K(2,2)$	$K(2,2)$			$f_{35} = x_1x_2 \vee x_3 \vee x_4 \vee x_5 \vee x_6$
	$K(2,2)$		$K(2,2)$	$K(1,2)$	$f_{36} = x_3 \vee x_4 \vee x_7 \vee x_8 \vee x_9x_{10}$
	$K(2,2)$	$K(1,2)$	$K(2,2)$		$f_{37} = x_3 \vee x_4 \vee x_5x_6 \vee x_7 \vee x_8$

$\{x_1, x_2\}$	$\{x_3, x_4\}$	$\{x_5, x_6\}$	$\{x_7, x_8\}$	$\{x_9, x_{10}\}$	
$K(1,2)$	$K(2,2)$		$K(2,2)$		$f_{38} = x_1x_2 \vee x_3 \vee x_4 \vee x_7 \vee x_8$
	$K(2,2)$		$K(1,2)$	$K(2,2)$	$f_{39} = x_3 \vee x_4 \vee x_7x_8 \vee x_9 \vee x_{10}$
	$K(2,2)$	$K(1,2)$		$K(2,2)$	$f_{40} = x_3 \vee x_4 \vee x_5x_6 \vee x_9 \vee x_{10}$
$K(1,2)$	$K(2,2)$			$K(2,2)$	$f_{41} = x_1x_2 \vee x_3 \vee x_4 \vee x_9 \vee x_{10}$
		$K(2,2)$	$K(2,2)$	$K(1,2)$	$f_{42} = x_5 \vee x_6 \vee x_7 \vee x_8 \vee x_9x_{10}$
	$K(1,2)$	$K(2,2)$	$K(2,2)$		$f_{43} = x_3x_4 \vee x_5 \vee x_6 \vee x_7 \vee x_8$
$K(1,2)$		$K(2,2)$	$K(2,2)$		$f_{44} = x_1x_2 \vee x_5 \vee x_6 \vee x_7 \vee x_8$
		$K(2,2)$	$K(1,2)$	$K(2,2)$	$f_{45} = x_5 \vee x_6 \vee x_7x_8 \vee x_9 \vee x_{10}$
	$K(1,2)$	$K(2,2)$		$K(2,2)$	$f_{46} = x_3x_4 \vee x_5 \vee x_6 \vee x_9 \vee x_{10}$
$K(1,2)$		$K(2,2)$		$K(2,2)$	$f_{47} = x_1x_2 \vee x_5 \vee x_6 \vee x_9 \vee x_{10}$
		$K(1,2)$	$K(2,2)$	$K(2,2)$	$f_{48} = x_5x_6 \vee x_7 \vee x_8 \vee x_9 \vee x_{10}$
	$K(1,2)$		$K(2,2)$	$K(2,2)$	$f_{49} = x_3x_4 \vee x_7 \vee x_8 \vee x_9 \vee x_{10}$
$K(1,2)$			$K(2,2)$	$K(2,2)$	$f_{50} = x_1x_2 \vee x_7 \vee x_8 \vee x_9 \vee x_{10}$

Отримався 50-ти реберний кільцевий граф.

2-*ий спосіб*

Можливі також і інші варіанти. Зокрема, можливо вихідну множину змінних розбити на 2 підмножини – на $\{x_1, x_2, x_3, x_4, x_5, x_6\}$ і $\{x_7, x_8, x_9, x_{10}\}$ відповідно. Комбінації різних $K(i, j)$, відповідні варіантам розподілу відмов по підмножинам наведено в табл. 2.4

Таблиця 2.4

$\{x_1, x_2, x_3, x_4, x_5\}$	$\{x_6, x_7, x_8, x_9, x_{10}\}$
$K(5, 5)$	
$K(4, 5)$	$K(1,5)$
$K(3, 5)$	$K(2, 5)$
$K(2, 5)$	$K(3, 5)$
$K(1,5)$	$K(4, 5)$

$\{x_1, x_2, x_3, x_4, x_5\}$	$\{x_6, x_7, x_8, x_9, x_{10}\}$
	$K(5, 5)$

Тепер розіб’єм кожну з підмножин змінних на свої підмножини: $\{x_1, x_2\}$, $\{x_3, x_4\}$, $\{x_5\}$, $\{x_6, x_7\}$, $\{x_8, x_9\}$ і $\{x_{10}\}$. Далі запишемо для кожного з $K(4, 5)$, $K(3, 5)$ $K(2, 5)$ свої таблиці комбінацій $K(2, 2)$ і $K(1, 2)$ (див. табл. 2.5, 2.6, 2.7, 2.8, 2.9, 2.10 відповідно).

Таблиця 2.5

$\{x_1, x_2\}$	$\{x_3, x_4\}$	$\{x_5\}$
$K(1, 2)$	$K(2, 2)$	$K(1, 1)$
$K(2, 2)$	$K(1, 2)$	$K(1, 1)$
$K(2, 2)$	$K(2, 2)$	

Таблиця 2.6

$\{x_1, x_2\}$	$\{x_3, x_4\}$	$\{x_5\}$
$K(1, 2)$	$K(1, 2)$	$K(1, 1)$
$K(2, 2)$	$K(1, 2)$	
$K(1, 2)$	$K(2, 2)$	
$K(2, 2)$		$K(1, 1)$
	$K(2, 2)$	$K(1, 1)$

Таблиця 2.7

$\{x_1, x_2\}$	$\{x_3, x_4\}$	$\{x_5\}$
$K(1, 2)$	$K(1, 2)$	
$K(1, 2)$		$K(1, 1)$
	$K(1, 2)$	$K(1, 1)$
$K(2, 2)$		
	$K(2, 2)$	

Таблиця 2.8

$\{x_6, x_7\}$	$\{x_8, x_9\}$	$\{x_{10}\}$
$K(1, 2)$	$K(2, 2)$	$K(1, 1)$
$K(2, 2)$	$K(1, 2)$	$K(1, 1)$
$K(2, 2)$	$K(2, 2)$	

Таблиця 2.9

$\{x_6, x_7\}$	$\{x_8, x_9\}$	$\{x_{10}\}$
$K(1, 2)$	$K(1, 2)$	$K(1, 1)$
$K(2, 2)$	$K(1, 2)$	
$K(1, 2)$	$K(2, 2)$	
$K(2, 2)$		$K(1, 1)$
	$K(2, 2)$	$K(1, 1)$

Таблиця 2.10

$\{x_6, x_7\}$	$\{x_8, x_9\}$	$\{x_{10}\}$
$K(1, 2)$	$K(1, 2)$	
$K(1, 2)$		$K(1, 1)$
	$K(1, 2)$	$K(1, 1)$
$K(2, 2)$		
	$K(2, 2)$	

Тепер, когли всі $K(i, j)$ задовольняють зазначеній вище, в першому способі формування, особливості, наважко записати реберні функції *GL*-моделі (не забуваючи про конкатенації).

$$f_1 = x_1 \vee x_2 \vee x_3 \vee x_4 \vee x_5$$

$$f_2 = x_1x_2 \vee x_3 \vee x_4 \vee x_5 \vee x_6x_7x_8x_9x_{10}$$

$$f_3 = x_1 \vee x_2 \vee x_3x_4 \vee x_5 \vee x_6x_7x_8x_9x_{10}$$

$$f_4 = x_1 \vee x_2 \vee x_3 \vee x_4 \vee x_6x_7x_8x_9x_{10}$$

$$f_5 = x_1x_2 \vee x_3x_4 \vee x_5 \vee x_6x_7 \vee x_8x_9$$

$f_6 = x_1x_2 \vee x_3x_4 \vee x_5 \vee x_6x_7 \vee x_{10}$

$f_7 = x_1x_2 \vee x_3x_4 \vee x_5 \vee x_8x_9 \vee x_{10}$

$f_8 = x_1x_2 \vee x_3x_4 \vee x_5 \vee x_6 \vee x_7$

$f_9 = x_1x_2 \vee x_3x_4 \vee x_5 \vee x_8 \vee x_9$

$f_{10} = x_1 \vee x_2 \vee x_3x_4 \vee x_6x_7 \vee x_8x_9$

$f_{11} = x_1 \vee x_2 \vee x_3x_4 \vee x_6x_7 \vee x_{10}$

$f_{12} = x_1 \vee x_2 \vee x_3x_4 \vee x_8x_9 \vee x_{10}$

$f_{13} = x_1 \vee x_2 \vee x_3x_4 \vee x_6 \vee x_7$

$f_{14} = x_1 \vee x_2 \vee x_3x_4 \vee x_8 \vee x_9$

$f_{15} = x_1x_2 \vee x_3 \vee x_4 \vee x_6x_7 \vee x_8x_9$

$f_{16} = x_1x_2 \vee x_3 \vee x_4 \vee x_6x_7 \vee x_{10}$

$f_{17} = x_1x_2 \vee x_3 \vee x_4 \vee x_8x_9 \vee x_{10}$

$f_{18} = x_1x_2 \vee x_3 \vee x_4 \vee x_6 \vee x_7$

$f_{19} = x_1x_2 \vee x_3 \vee x_4 \vee x_8 \vee x_9$

$f_{20} = x_1 \vee x_2 \vee x_5 \vee x_6x_7 \vee x_8x_9$

$f_{21} = x_1 \vee x_2 \vee x_5 \vee x_6x_7 \vee x_{10}$

$f_{22} = x_1 \vee x_2 \vee x_5 \vee x_8x_9 \vee x_{10}$

$f_{23} = x_1 \vee x_2 \vee x_5 \vee x_6 \vee x_7$

$f_{24} = x_1 \vee x_2 \vee x_5 \vee x_8 \vee x_9$

$f_{25} = x_3 \vee x_4 \vee x_5 \vee x_6x_7 \vee x_8x_9$

$f_{26} = x_3 \vee x_4 \vee x_5 \vee x_6x_7 \vee x_{10}$

$f_{27} = x_3 \vee x_4 \vee x_5 \vee x_8x_9 \vee x_{10}$

$f_{28} = x_3 \vee x_4 \vee x_5 \vee x_6 \vee x_7$

$f_{29} = x_3 \vee x_4 \vee x_5 \vee x_8 \vee x_9$

$f_{30} = x_6x_7 \vee x_8x_9 \vee x_{10} \vee x_1x_2 \vee x_3x_4$

$f_{31} = x_6x_7 \vee x_8x_9 \vee x_{10} \vee x_1x_2 \vee x_5$

$f_{32} = x_6x_7 \vee x_8x_9 \vee x_{10} \vee x_3x_4 \vee x_5$

$f_{33} = x_6x_7 \vee x_8x_9 \vee x_{10} \vee x_1 \vee x_2$

$f_{34} = x_6x_7 \vee x_8x_9 \vee x_{10} \vee x_3 \vee x_4$

$f_{35} = x_6 \vee x_7 \vee x_8 \vee x_9 \vee x_1x_2 \vee x_3x_4$

$f_{36} = x_6 \vee x_7 \vee x_8 \vee x_9 \vee x_1x_2 \vee x_5$

$f_{37} = x_6 \vee x_7 \vee x_8 \vee x_9 \vee x_3x_4 \vee x_5$

$f_{38} = x_6 \vee x_7 \vee x_8 \vee x_9 \vee x_1 \vee x_2$

$f_{39} = x_6 \vee x_7 \vee x_8 \vee x_9 \vee x_3 \vee x_4$

$f_{40} = x_6x_7 \vee x_8x_9 \vee x_1x_2 \vee x_3x_4$

$f_{41} = x_6x_7 \vee x_8x_9 \vee x_1x_2 \vee x_5$

$f_{42} = x_6x_7 \vee x_8x_9 \vee x_3x_4 \vee x_5$

$f_{43} = x_6x_7 \vee x_8x_9 \vee x_1 \vee x_2$

$f_{44} = x_6x_7 \vee x_8x_9 \vee x_3 \vee x_4$

$f_{45} = x_6 \vee x_7 \vee x_{10} \vee x_1x_2 \vee x_3x_4$

$f_{46} = x_6 \vee x_7 \vee x_{10} \vee x_1x_2 \vee x_5$

$f_{47} = x_6 \vee x_7 \vee x_{10} \vee x_3x_4 \vee x_5$

$f_{48} = x_6 \vee x_7 \vee x_{10} \vee x_1 \vee x_2$

$f_{49} = x_6 \vee x_7 \vee x_{10} \vee x_3 \vee x_4$

$f_{50} = x_8 \vee x_9 \vee x_{10} \vee x_1x_2 \vee x_3x_4$

$f_{51} = x_8 \vee x_9 \vee x_{10} \vee x_1x_2 \vee x_5$

$f_{52} = x_8 \vee x_9 \vee x_{10} \vee x_3x_4 \vee x_5$

$f_{53} = x_8 \vee x_9 \vee x_{10} \vee x_1 \vee x_2$

$f_{54} = x_8 \vee x_9 \vee x_{10} \vee x_3 \vee x_4$

$f_{55} = x_1x_2x_3x_4x_5 \vee x_6x_7 \vee x_8x_9 \vee x_{10}$

$f_{56} = x_1x_2x_3x_4x_5 \vee x_6 \vee x_7 \vee x_8x_9 \vee x_{10}$

$f_{57} = x_1x_2x_3x_4x_5 \vee x_6 \vee x_7 \vee x_8 \vee x_9$

$f_{58} = x_6 \vee x_7 \vee x_8 \vee x_9 \vee x_{10}$

Як бачимо, в обох випадках число функцій (відповідно, ребер графа) більше числа модулів ВБС, проте функції не тільки безповторні, але й іноді не залежать від змінних, що полегшує роботу з ними.

2.4 <u>Спрощені способи побудови *GL*-моделей для базових *K*(*m*, *n*)-систем при *m* > 2</u>

У розглянутому вище способі побудови GL-моделей виходить досить велике число реберних функцій, і, відповідно, ребер графа. Це не зручно, особливо при великих значеннях n. Тому були розроблені різні варіанти оптимізації описаних вище принципів побудови моделі, які дозволяють створити *GL*-модель з меншою кількістю ребер. Розглянемо один з них на прикладі побудови K(5, 10).

1. Розбиваємо вхідну множину змінних, що позначають стан елементів системи, на 2 підмножини рівної потужності при парній кількості елементів системи, а при непарному – на множини з потужностями $\left[\frac{n}{2}\right]$ і $\left[\frac{n}{2}\right]+1$ відповідно (запис $\left[\frac{n}{2}\right]$ означає округлення у бік меншого цілого числа).
2. Будуємо таблицю, яка містить у собі варіанти можливого розподілу відмов по підсистемам:

Таблиця 2.11

$\{x_1, x_2, \ldots, x_{\left[\frac{n}{2}\right]}\}$	$\{x_{\left[\frac{n}{2}\right]+1}, x_{\left[\frac{n}{2}\right]+2}, \ldots, x_n\}$
$K(m, \left[\frac{n}{2}\right])$	
$K(m–1, \left[\frac{n}{2}\right])$	$K(1, n – \left[\frac{n}{2}\right])$
$K(m–2, \left[\frac{n}{2}\right])$	$K(2, n – \left[\frac{n}{2}\right])$
…	…
	$K(m, n – \left[\frac{n}{2}\right])$

3. Далі розглядаємо кожнен рядок з отриманої таблиці. Зробимо так, щоб одному рядку відповідала одна реберна функція. Для кожної з $K(m, \left[\frac{n}{2}\right])$, $K(m–1, \left[\frac{n}{2}\right])$, $K(m–2, \left[\frac{n}{2}\right])$, …, $K(1, \left[\frac{n}{2}\right])$, $K(1, n – \left[\frac{n}{2}\right])$, $K(2, n – \left[\frac{n}{2}\right])$, …, $K(m, n – \left[\frac{n}{2}\right])$ виписуєм в таблиці можливі комбінації з $K(2, 2)$ і $K(1, 2)$.
4. Тепер, коли всі $K(i, j)$ відповідають особливостям (2.5) і (2.6), запишемо всі функції, отримані для відповідної $K(i, j)$ і об'єднані конкатенації. В результаті для кожної $K(i, j)$ з першої таблиці вийде одна функція.
5. Запишемо тепер реберні функції *GL*-моделі *K(m, n)* системи. Для цього необхідно взяти диз'юнкції функцій, що стоять в одному рядку підсистем, тобто функцій, одержаних для пар $K(m–1, \left[\frac{n}{2}\right])$ і $K(1, n – \left[\frac{n}{2}\right])$, $K(m–2, \left[\frac{n}{2}\right])$ і $K(2, n – \left[\frac{n}{2}\right])$, …, $K(1, \left[\frac{n}{2}\right])$ і

$K(m-1\ ,\ n-\left\lceil \frac{n}{2} \right\rceil)$. Таким чином, отримується всього $m+1$ реберна функція, і, відповідно, $(m+1)$-реберний кільцевий граф.

Розглянемо докладніше цей спосіб на прикладі. Побудуємо *GL*-модель для базової $K(5,\ 10)$ системи. Розіб'ємо вхідну множину змінних на 2 підмножини $\{x_1,\ x_2,\ x_3,\ x_4,\ x_5\}$ $\{x_6,\ x_7,\ x_8,\ x_9,\ x_{10}\}$ і скористаємося вже побудованими таблицями 2.4, 2.5, 2.6, 2.8, 2.9 та 2.10 з розподілом можливих варіантів відмов по підсистемах. Отримаємо наступний результат:

$f_1 = x_1 \vee x_2 \vee x_3 \vee x_4 \vee x_5$

$f_2 = (x_1x_2 \vee x_3 \vee x_4 \vee x_5)(x_1 \vee x_2 \vee x_3x_4 \vee x_5)(x_1 \vee x_2 \vee x_3 \vee x_4) \vee x_6x_7x_8x_9x_{10}$

$f_3 = (x_1x_2 \vee x_3x_4 \vee x_5)(x_1 \vee x_2 \vee x_3x_4)(x_1x_2 \vee x_3 \vee x_4)(x_1 \vee x_2 \vee x_5)\ \&$

$\&\ (x_3 \vee x_4 \vee x_5) \vee (x_6x_7 \vee x_8x_9)(\ x_6x_7 \vee x_{10})(x_8x_9 \vee x_{10})(x_6 \vee x_7)(x_8 \vee x_9)$

$f_4 = (x_6x_7 \vee x_8x_9 \vee x_{10})(x_6 \vee x_7 \vee x_8 \vee x_9)(x_6x_7 \vee x_8x_9)(x_6 \vee x_7 \vee x_{10})\ \&$

$\&\ (x_8 \vee x_9 \vee x_{10}) \vee (x_1x_2 \vee x_3x_4)(x_1x_2 \vee x_5)(x_3x_4 \vee x_5)(x_1 \vee x_2)(x_3 \vee x_4)$

$f_5 = x_1x_2x_3x_4x_5 \vee (x_6x_7 \vee x_8x_9 \vee x_{10})(x_6 \vee x_7 \vee x_8x_9 \vee x_{10})(x_6 \vee x_7 \vee x_8 \vee x_9)$

$f_6 = x_6 \vee x_7 \vee x_8 \vee x_9 \vee x_{10}$

Цьому набору функцій буде відповідати 6-реберний кільцевої граф, чию зв'язність аналізувати значно легше. Крім того, цей граф буде значно простіше модифікувати у разі можливого переходу до небазової системі.

3. ВІДОМІ МЕТОДИ РОЗРАХУНКУ НАДІЙНОСТІ БАЗОВИХ ВБС

3.1 Структурна функція *k*-out-of-*n*:G и *k*-out-of-*n*:F системи.

Останнім часом з'явилося багато робіт, особливо зарубіжних авторів, присвячених так званим k-out-of-n (або «k з n») системам - дослідженням їх властивостей, розробці різних алгоритмів оптимізації процесу визначення стану системи, пропонуються формули розрахунку надійності характеристик [14, 15]. K-out-of n система працездатна, якщо k модулів з n функціонують, тобто в нашому визначенні це - базова система. З одного боку, цей інтерес підкреслює теоретичну і практичну важливість проблеми, а з іншого - її складність, тому що базові ВБС є найпростішими. При цьому, як правило, передбачається наявність будь-яких обмежень, що полегшують рішення задачі. Розглянемо деякі з методик розрахунку надійності, запропонованих зарубіжними фахівцями, відзначаючи їх переваги та недоліки.

Відзначимо, що існує розбиття k-out-of-n систем на два типи:

k-out-of-n: G і k-out-of-n: F системи.

Система з n компонентів, що працює (або вважається справною) тоді і тільки тоді, коли хочаб працюють (або вважаються справними), k з n її компонентів називається k-out-of-n: G системою. Система, що складається з n компонентів і відмовляє тоді і тільки тоді, коли відмовляють хоча б k з n її компонентів, називається k-out-of-n: F системою. Виходячи з цих двох визначень, k-out-of-n: G система є еквівалентною до (n - k + 1) -out-of-n: F системі. термін«K-out-of-n система» часто вживають, щоб позначити або F систему, або G систем, або обидві ці системи. Паралельні і послідовні системи є окремими випадками k-out-of-n систем. Послідовна система є еквівалентом 1-out-of-n: F системи і n-out-of-n: G системи. Паралельна ж система - еквівалент n-out-of-n: F системи і 1-out-of-n: G системи.

Для оцінки надійності k-out-of-n: G систем, компоненти яких необов'язково мають рівну надійність, можна застосувати метод мінімального набору шляхів і розрізів. Одними з найбільш ефективних алгоритмів для оцінки надійності таких систем є алгоритми, запропоновані Барлоу і Хейдтманном (Barlow and Heidtmann), а також - Рушді (Rushdi). Обчислювальна складність цих двох алгоритмів однакова, порядку O (k (n-k + 1)). Ці методи будуть розглянуті в даному розділі.

Всі ці методи грунтуються на використанні такої математичної моделі системи, як логічна структурна функція системи. F (X), де логічний вектор $x = (x_1, x_2, \ldots, x_n)$ характеризує працездатність елементів системи.

Нехай $x_i = 1$ означає, що i-й елемент системи (підсистема) працездатний, а $x_i = 0$, що i-й елемент відмовив, а функція $F(X)$ обрана так, що вона дорівнює одиниці тоді і тільки тоді, коли система працездатна; вважається що $x \geq y$, якщо $x_i \geq y_i$ для усіх $i = \overline{1, n}$.

Два компонента називаються симетричними, якщо їх взаємна перестановка в системі не впливає на стан системи. Тобто, компоненти i та j є симетричними, якщо для всіх векторів стану системи виконується така умова:

$$\varphi(x_1, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_{j-1}, x_j, x_j+1, \ldots, x_n) =$$

$$= \varphi(x_1, \ldots, x_{i-1}, x_j, x_{i+1}, \ldots, x_{j-1}, x_i, x_{j+1}, \ldots, x_n).$$

Структурна функція пов'язаної системи називається симетричною, якщо всі її компоненти симетричні між собою. Симетричні системи включають в себе послідовні, паралельні і k-out-of-n структури.

3.2 Метод мінімального набору шляхів і розрізів

Цей метод є наближеним, що дозволяє оцінити дійсне значення ймовірності безвідмовної роботи системи зверху і знизу, він є більш простим, ніж відомі точні методи і тому підходить для оцінки надійності досить складних систем.

Для формального опису методу використовується логічна структурна функція системи.

Визначимо в прийнятих позначеннях поняття «мінімальний шлях» і «мінімальний розріз».

Якщо F (X) = 1 і F (Y) = 0 при будь-яких $y < x$, тоді $x = \alpha$ – мінімальний шлях, тобто,j-й мінімальний шлях складається з локально мінімальної сукупності M_j підсистем, необхідної для забезпечення безвідмовної роботи системи незалежно від стану інших підсистем. У структурі системи, як правило, є кілька мінімальних шляхів. Характерною ознакою мінімального шляху є те, що відмова хоча б однієї підсистеми шляху (якщо працездатні тільки підсистеми шляху) призводить до відмови системи в цілому.

Якщо F (X) = 0 і F (Y) = 1 при будь-якому y> x, то x = β - мінімальний переріз, тобто k-е мінімальний переріз складається з мінімальної сукупності підсистем Nk, одночасну відмову яких тягне за собою відмову системи незалежно від стану інших підсистем. Характерною особливістю мінімального розрізу є те, що відновлення хоча б однієї підсистеми в мінімальному розрізі (якщо всі інші підсистеми працездатні) тягне за собою відновлення системи.

За методом мінімальних шляхів і розрізів можна отримати тільки оцінки $p_н$ та $p_в$ ймовірності безвідмовної роботи системи відповідно знизу і зверху. Імовірність безвідмовної роботи системи оцінюється тоді за подвійним нерівності:

$p_н \leq p_c \leq p_в$.

Ймовірність $p_н$ виражається як ймовірність безвідмовної роботи допоміжної системи, складеної з послідовно включених груп підсистем, які відповідають усім мінімальним перетинах системи, кожна група складається з паралельно включених підсистем відповідного мінімального розрізу.

Ймовірність $p_в$ виражається як ймовірність безвідмовної роботи допоміжної системи, складеної з паралельно включених груп підсистем, які відповідають усім мінімальним шляхах системи. Кожна група складається з послідовно включених підсистем відповідного мінімального шляху

Як випливає зі сказаного, метод мінімальних шляхів і мінімальних розрізів дозволяє звести аналіз будь-яких систем до аналізу систем з послідовно-паралельної і паралельно-послідовною структурою і тому може бути використаний для аналізу досить складних систем при помірної складності одержуваних формул. Недоліком методу є те, що рішення виходить наближене, у вигляді оцінок знизу і зверху

[16, 23].

Як було зазначено раніше, надійність будь-якої системи дорівнює ймовірності того, що працює хоча б один з мінімальних наборів шляхів. Ненадійність ж системи дорівнює ймовірності того що, по крайней мере, один мінімальний розріз відмовив. Для того, що б мінімальний набір шляхів працював, всі його компоненти повинні працювати. Для того щоб відмовив мінімальний набір розрізів, необхідно щоб усі його компоненти відмовили. В k-out-of-n: G системах є $\binom{n}{k}$ мінімальних наборів шляхів і $\binom{n}{n-k+1}$ мінімальних наборів розрізів. Кожен мінімальний набір шляхів містить рівно k різних компонент, а кожен мінімальний набір розрізів - рівно (n - k + 1) різних компонент. Т.ч. всі мінімальні набори шляхів і розрізів відомі. Залишається відкритим питання, як визначити ймовірність того, що, по крайней мере, в одному з мінімальних наборів шляхів все компоненти справні, або ж ймовірність того, що, по крайней мере, в одному з мінімальних наборів розрізів всі компоненти відмовили.

Для оцінки надійності k-out-of-n: G систем, в яких відомі всі мінімальні набори шляхів і розрізів, можна застосувати метод ВВ (включення - виключення). Недолік цього методу полягає в тому, що він призводить до великої кількості скорочуються змінних, що негативно впливає на точність. Хейдтманн і МакГрейд (McGrady) запропонували поліпшені варіанти методу BI для оцінки надійності

k-out-of-n: G систем. У цих поліпшених алгоритмах виключається скорочення змінних. Однак обидва ці алгоритму за своєю суттю є алгоритмами перебору. Наприклад, у формулі, запропонованої Хейдтманном, мінімальні набори шляхів використовуються наступним чином:

$$P(k,n) = \sum_{i=k}^{n} (-1)^{i-k} \binom{i-1}{k-1} \sum_{j_1 < j_2 < \ldots < ji} \prod_{l=1}^{i} p_{j_l}\ , \tag{3.1}$$

где *P (k, n) - надійність k-out-of-n: G системи.*

У цьому виразі для кожного фіксованого значення i, внутрішній член суми дає нам можливість того, що i-й компонент справний незалежно від того, чи справні інші (n - i) компонента. Загальна кількість членів у внутрішній сумі ряду одно $\binom{n}{i}$. Якщо всі компоненти системи є незалежними і мають однакові функції розподілу безвідмовної роботи (independent and identically distributed, i.i.d.), вираз (3.1) дасть нам наступну формулу для оцінки надійності k-out-of-n: G системи з i.i.д компонентами :

$$P(k,n) = \sum_{i=k}^{n} \binom{n}{i} \binom{i-1}{k-1} (-1)^{i-k} p^i \tag{3.2}$$

Також, для оцінки надійності k-out-of-n: G систем може бути застосований метод СНД (суми непересікаючихся добутків). Подібно поліпшеному методу ВВ, він легко застосовний до k-out-of-n: G системам. Але існують методи оцінки надійності k-out-of-n: G систем, набагато ефективніші, ніж методи ВВ (і його покращений варіант) і СНД. Далі наводиться приклад, який ілюструє використання мінімальних наборів шляхів методом ВВ, поліпшеним Хейдтманном методом ВВ і методом СНД для оцінки надійності 2-out-of-4: G системи.

Приклад Оцінити надійність 2-out-of-4: G системи, якщо $p_1 = 0.91$, $p_2 = 0.92$, $p_3 = 0.93$, і $p_4 = 0.94$. Кількість мінімальних шляхів дорівнює $\binom{4}{2} = 6$. Позначення S_i будемо використовувати для подання i-го мінімального шляху, так як вказано нижче:

$$S_1 = x_1x_2, \qquad S_2 = x_1x_3, \qquad S_3 = x_1x_4,$$

$$S_4 = x_2x_3, \qquad S_5 = x_2x_4, \qquad S_6 = x_3x_4.$$

Використовуючи метод ВВ, ми можемо обчислити надійності системи в такий спосіб:

$$\begin{aligned}
P(2, 4) &= Pr(S_1 \cup S_2 \cup S_3 \cup S_4 \cup S_5 \cup S_6) \\
&= Pr(S_1) + Pr(S_2) + Pr(S_3) + Pr(S_4) + Pr(S_5) + Pr(S_6) - \\
&-Pr(S_1S_2) - Pr(S_1S_3) - Pr(S_1S_4) - Pr(S_1S_5) - Pr(S_1S_6) - Pr(S_2S_3) - \\
&-Pr(S_2S_4) - Pr(S_2S_5) - Pr(S_2S_6) - Pr(S_3S_4) - Pr(S_3S_5) - Pr(S_3S_6) - \\
&-Pr(S_4S_5) - Pr(S_4S_6) - Pr(S_5S_6) + Pr(S_1S_2S_3) + Pr(S_1S_2S_4) + \\
&+ Pr(S_1S_2S_5) + Pr(S_1S_2S_6) + Pr(S_1S_3S_4) + Pr(S_1S_3S_5) + Pr(S_1S_3S_6) + \\
&+Pr(S_1S_4S_5) + Pr(S_1S_4S_6) + Pr(S_1S_5S_6) + Pr(S_2S_3S_4) + Pr(S_2S_3S_5) + \\
&+Pr(S_2S_3S_6) + Pr(S_2S_4S_5) + Pr(S_2S_4S_6) + Pr(S_2S_5S_6) + Pr(S_3S_4S_5) + \\
&+Pr(S_3S_4S_6) + Pr(S_3S_5S_6) + Pr(S_4S_5S_6) - Pr(S_1S_2S_3S_4) - Pr(S_1S_2S_3S_5) \\
&- Pr(S_1S_2S_3S_6) - Pr(S_1S_2S_4S_5) - Pr(S_1S_2S_4S_6) - Pr(S_1S_2S_5S_6) - \\
&-Pr(S_2S_3S_4S_5) - Pr(S_2S_3S_4S_6) - Pr(S_2S_3S_5S_6) - Pr(S_2S_4S_5S_6) - \\
&-Pr(S_3S_4S_5S_6) + Pr(S_1S_2S_3S_4S_5) + Pr(S_1S_2S_3S_4S_6) + Pr(S_1S_2S_3S_5S_6) + \\
&+Pr(S_1S_2S_4S_5S_6) + Pr(S_1S_3S_4S_5S_6) + Pr(S_2S_3S_4S_5S_6) \approx 0{,}998467.
\end{aligned}$$

Використовуючи вираз (3.1), отримуємо

$$P(2,4) = \sum_{i=2}^{4}(-1)^{i-2}\binom{i-1}{1}\sum_{j_1<j_2<\cdots<j_i}\prod_{l=1}^{i}p_{j_l} =$$

$= (p_1p_2 + p_1p_3 + p_1p_4 + p_2p_3 + p_3p_4) – 2(p_1p_2p_3 + p_1p_2p_4 + p_1p_3p_4 + p_2p_3p_4) +$

$+ 3p_1p_2p_3p_4 \approx 0{,}998441.$

Використовуючи метод СНП, отримуємо

$P(2, 4) = Pr(S_1 \cup S_2 \cup S_3 \cup S_4 \cup S_5 \cup S_6) =$

$= Pr(S_1) + Pr(\overline{S_1}\,S_2) + Pr(\overline{S_1}\,\overline{S_2}\,S_3) + Pr(\overline{S_1}\,\overline{S_2}\,\overline{S_3}\,S_4) +$

$+ Pr(\overline{S_1}\,\overline{S_2}\,\overline{S_3}\,\overline{S_4}\,S_5) + Pr(\overline{S_1}\,\overline{S_2}\,\overline{S_3}\,\overline{S_4}\,\overline{S_5}\,S_6) =$

$= Pr(x_1x_2) + Pr(x_1\overline{x_2}\,x_3) + Pr(x_1\overline{x_2}\,\overline{x_3}\,x_4) + Pr(\overline{x_1}\,x_2x_3) + Pr(\overline{x_1}\,x_2\overline{x_3}\,x_4) +$

$+ Pr(\overline{x_1}\;\overline{x_2}\,x_3x_4) \approx 0{,}998441.$

Очевидно, що метод ВВ вимагає набагато більше обчислень, ніж оптимізований ВВ метод і метод СНД. Також очевидно, що велика кількість скорочуються змінних в методі ВВ призводить до помилок округлення кінцевого результату. Основним недоліком обох цих методів, пов'язаних з мінімальними шляхами і мінімальними розрізами є те, що вони можуть бути застосовані тільки для оцінки надійності систем з невеликим числом компонент.

3.3 Метод Барлоу і Хейдтманна

Барлоу і Хейдтманн представили два алгоритму для оцінки надійності систем з незалежними компонентами. Перший алгоритм використовує наступну функцію генерування і її розширену форму:

$$g_n(z) = \prod_{i=0}^{n}(q_i + p_i z) = \sum_{i=0}^{n}P_e(i,n)z^i, \tag{3.3}$$

де z – фіктивна змінна, а $P_e(i, n)$ – ймовірність того, що в системі працює рівно i компонент. З розширеної форми $g_n(z)$ можна відзначити, що $P_e(i, n)$

також представляє коефіцієнт при z^i в функції генерації. Відповідно до алгоритму всі значення $P_e(i, n)$ обчислюються рекурсивно. Рушді запропонував краще пояснення цього алгоритму. По суті, алгоритм грунтується на формулі

$$P(k,n) = \sum_{i=k}^{n} P_e(i,n), \tag{3.4}$$

яка безпосередньо випливає з визначення k-out-of-n: G системи. Алгоритм отримує значення $P_e(i, n)$ з рекурсивного співвідношення

$$P_e(i, j) = q_j P_e(i, j-1) + p_j P_e(i-1, j-1), \qquad 0 \le i \le n, \qquad 0 \le j \le n, \tag{3.5}$$

з граничними умовами

$$P_e(-1, j) = P_e(j+1, j) = 0 \qquad \text{для } j = 0, 1, 2, \ldots. \tag{3.6}$$

$$P_e(0, 0) = 1. \tag{3.7}$$

Для того щоб перейти до цього рекурсивному відношенню, необхідно скласти функцію генерування:

$$g_{j-1}(z) = \prod_{i=1}^{j-1} (q_i + p_i z) = \sum_{i=0}^{j-1} P_e(i,\ j-1) z^i. \tag{3.8}$$

Так як $g_j(z) = (q_j + p_j z) g_{j-1}(z)$, то порівняння коефіцієнтів при z^i в обох частинах виразу

$$\sum_{i=0}^{j} P_e(i, j) z^i = (q_i + p_j z) \sum_{i=0}^{j-1} P_e(i, j-1) z^i = \sum_{i=0}^{j} [q_j P_e(i, j-1) + p_j P_e(i-1, j-1)] z^i \tag{3.9}$$

призводить до вираження (3.5).

Для того щоб визначити обчислювальну складність алгоритму, було досліджено кількість значень $P_e(i, j)$, які повинні бути обчислені за формулою (3.5), використовуючи граничні умови (3.6) і (3.7). Як показано в таблиці 3.1, загальна кількість значень $P_e(i, j)$ дорівнює

$$(n - k + 1)(k + 1) - 1 + ½\,(n - k)^2.$$

Обчислення кожного такого значення вимагає трьох основних арифметичних операцій (двох операцій додавання і однієї - множення). Потім, для того щоб знайти надійність системи, нам доведеться використовувати вираз (3.4), обчислення якого вимагає n - k основних арифметичних операцій. Тобто загальне число основних арифметичних операцій, необхідних для обчислення надійності системи по данному алгоритму, так само

$$3[(n - k + 1)(k + 1) - ½\,(n - k)^2] + n - k = (n - k)(1{,}5n + 1{,}5k + 4) + 3k.$$

З цього виразу можна зробити висновок, що обчислювальна складність алгоритму порядку $O(n^2)$ при малих значеннях k (які прагнуть до 1) і порядку $O(n)$ при великих значеннях k (які прагнуть до n). У загальному випадку, вважається, що обчислювальна складність даного алгоритму порядку $O(n^2)$.

Кількість арифметичних операцій, необхідних для оцінки надійності системи, може бути зменшено, якщо відзначити той факт, що нас цікавить ймовірність роботи, по крайней мере, k компонент системи. Т.ч. обчислень $P_e(i, j)$ можна уникнути. Другий алгоритм Барлоу і Хейдтманна не вирахував $P_e(i, j)$ і вимагає лише 3 (n - k +1) арифметичних операцій. Ця ж обчислювальна складність досягається алгоритмом, запропонованим Рушді і описаним далі.

j \ i	0	1	2	3		k	$k+1$		$n-k$	$n-k+1$	$n-k+2$				n
-1	0	0	0	0		0	0		0						
0	1	$(0,1)$	$(0,2)$	$(0,3)$		$(0, k)$	$(0, k+1)$		$(0, n-k)$	$(0, n-k+1)$					
1	0	$(1,1)$	$(1,2)$	$(1,3)$		$(1, k)$	$(1, k+1)$		$(1, n-k)$	$(1, n-k+1)$	$(1, n-k+2)$				
2		0	$(2,2)$	$(2,3)$		$(2, k)$	$(2, k+1)$		$(2, n-k)$	$(2, n-k+1)$	$(2, n-k+2)$				
…			0	$(3,3)$		…	…		…	…	…				
…				0		…	…		…	…	…				
k						(k, k)	$(k, k+1)$		$(k, n-k)$	$(k, n-k+1)$	$(k, n-k+2)$				(k, n)
$k+1$							$(k+1, k+1)$		$(k+1, n-k)$		…				$(k+1, n)$
…									…		…				…
$n-k$									$(n-k, n-k)$		…				$(n-k, n)$
$n-k+1$									0	$(n-k+1, n-k+1)$	…				$(n-k+1, n)$
$n-k+2$										0	…				$(n-k+2, n)$
…											0				
…															
…															
n															(n, n)

Таблиця 3.1 Значення $P_e(i, j)$, які повинні бути обчислені згідно з алгоритмом Барлоу і Хейдтманна

3.4 Метод Рушді

Цей метод починається з аналізу структурної функції k-out-of-n: G системи. Структурна функція $\varphi(x)$ k-out-of-n:G системи, за визначенням, наведеним вище, є симетричною. Вона може приймати тільки два значення (0 і 1), в зв'язку з двійковим поданням стану компонент і системи в цілому, подібно перемикачу. Ось чому цей метод називається методом симетричною перемикальної функції.

Нехай xi означає стан елемента i, а S (k, n), замість $\varphi(x)$ означає структурну функцію системи. І x_i, і $S(k, n)$ є двійковими змінними, середнє арифметичне значення яких означає робочий стан, а нульове - відмова. Інверсія цих змінних представляється як $\overline{x_i}$ і $\overline{S}(k,n)$ відповідно. Грунтуючись на цих визначеннях $S(k, n)$ і $\overline{S}(k,n)$, маємо такі вирази для надійності системи і її ненадійності:

$$P(k, n) = \Pr(S(k, n) = 1), \qquad Q(k, n) = \Pr(\overline{S}(k,n) = 1).$$

Для того щоб знайти вираз, що визначає стан системи, можна застосувати основне (центральне) розкладання по n-му компоненту, так як це показано нижче:

$$S(k, n) = x_n(S(k-1, n-1)) + \overline{x_n}\,(S(k, n-1)), \tag{3.10}$$

$$\overline{S}(k,n) = x_n(\overline{S}(k-1,n-1)) + \overline{x_n}\,(\overline{S}(k,n-1)). \tag{3.11}$$

Грунтуючись на цих двох рівняннях, стан k-out-of-n: G системи може бути виражено як функція з станами двох підсистем з такими ж n - 1 компонентами. Проте, мінімальна кількість компонент, необхідний для того, щоб ці дві підсистеми працювали по-різному. Однією необхідно, по крайней мере, k справних компонент для роботи, а інший - k - 1 компонент. Далі, ці дві підсистеми з n - 1 компонентами можуть бути розкладені за останнім компоненту, а саме - по (n - 1) -го до того, як будуть досягнуті деякі граничні умови. Таким чином, для опису процесу розкладання може бути використано итеративне вираз. Уявімо собі систему, що складається з j компонент, для роботи якої необхідно, щоб як мінімум і компонент були справні. У нас є такі рівняння для вираження стану такої системи у вигляді функції станів двох підсистем:

$$S(i, j) = x_j(S(i–1, j–1)) + \overline{x_j}\,(S(i, j–1)), \tag{3.12}$$

$$\overline{S}(i, j) = x_j(\overline{S}(j-1, i-1)) + \overline{x_j}\,(\overline{S}(i, j-1)), \tag{3.13}$$

де і може приймати будь-яке цілочисельне значення від 1 до k, а j - може приймати значення від 0 до n. Наступні граничні умови необхідні для рівнянь (3.12) і (3.13):

$$S(0, j) = \overline{S}\,(j+1, j) = 1, \tag{3.14}$$

$$S(j+1, j) = \overline{S}\,(0, j) = 0. \tag{3.15}$$

Рівняння (3.12) і (3.13) записані у формі основного (центрального) розкладання. У зв'язку з допущенням того, що компоненти є незалежними, вони можуть бути перетворені до наступних алгебраїчних виразів надійності:

$$P(i, j) = p_jP(i – 1, j – 1) + q_jP(i, j – 1), \tag{3.16}$$

$$Q(i, j) = p_jQ(i – 1, j – 1) + q_jQ(i, j – 1). \tag{3.17}$$

Рівняння (3.16) і (3.17) є рекурсивними співвідношеннями, допустимими при $1 \le i \le k$. Граничні умови для них можуть бути отримані безпосередньо з рівнянь (3.14) і (3.15) в наступному вигляді:

$$P(0, j) = Q(j+1, j) = 1, \tag{3.18}$$

$$P(j+1, j) = Q(0, j) = 0. \tag{3.19}$$

Рішення для розрахунку надійності P (k, n) або для ненадійності Q (k, n) легко можуть бути отримані програмуванням на тих мовах, які дозволяють рекурсивні виклики. Однак, більш детальний розгляд рекурсивних співвідношень (3.16) і (3.17) виявляє те, що вони без особливих зусиль можуть бути представлені сигнальним орієнтованим графом (СОГ). Як приклад на малюнку 3.1 показаний СОГ для обчислення P (3, 7). На рис. 3.1

вузол на позиції (i, j) представляє $P(i, j)$. Чорні вузли в першому ряду з і = 0 є «вихідними» вузлами зі значеннями 1, що означає $P(0, j) = 1$. Білі вузли, у яких $i = j + 1$, є «вихідними» вузлами з нульовими значеннями, тобто

$P(j+1,j)=0$ для $j \geq 0$. Значення в інших вузлах, наприклад в (i, j), повинні бути обчислені за допомогою складання твори найближчого верхнього зліва значення і p_j з добутком найближчого лівого значення і q_j.

Той же граф на малюнку 3.1 може бути використаний для обчислення Q (3, 7), якщо вважати, що вузол (i, j) представляє ненадійність $Q(i, j)$, замість надійності $P(i, j)$, а два типу вихідних вузлів змінюються своїми значеннями. Це означає, що чорні вузли, у яких і = 0 стають нульовими значеннями

[$Q(0, j) = 0$], а білі вузли, у яких $i = j + 1$, стають поодинокими вузлами [$Q(j+1, j) = 1$].

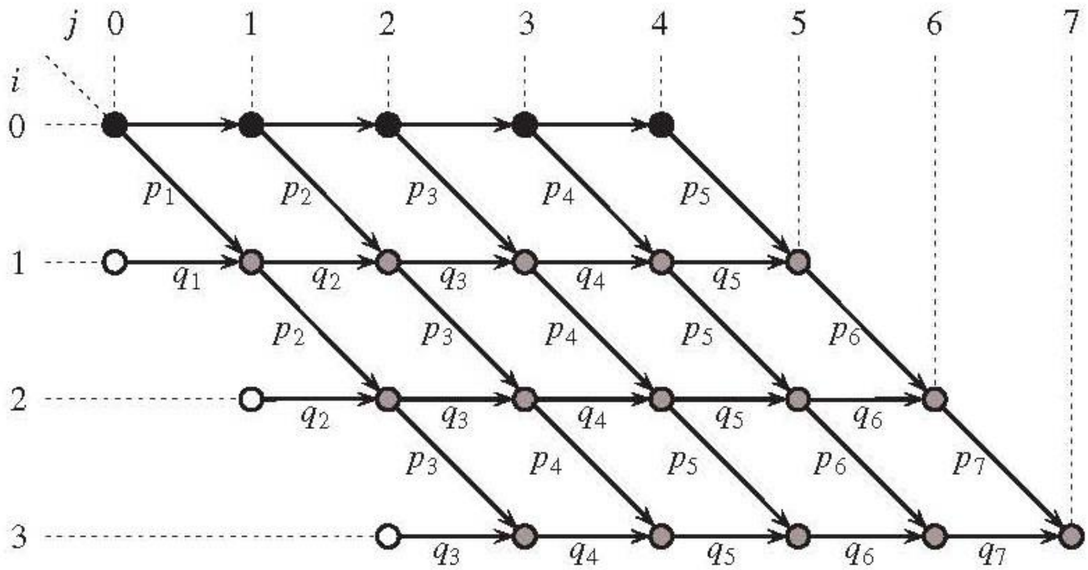


Рисунок. 3.1 - Сигнальний орієнтований граф для отримання $P(3, 7)$і

$Q(3, 7)$

Алгоритм є ефективним за рахунок прямого побудови (тобто обчислення значень елементів) паралелограма з кутами (1,1),

(1, n - k +1), (k, k) і (k, n). Кількість елементів в параллелограмме дорівнює k (n - k + 1). Обчислення кожного елемента паралелограма вимагає виконання трьох арифметичних операцій, а саме - одного множення і двох додавань. Це неважко побачити, якщо спростити вирази (3.16) і (3.17), підставивши в них співвідношення qj = 1 - pj наступним чином:

$$P(i, j) = p_j P(i-1, j-1) + (1-p_j)P(i, j-1) =$$
$$= P(i, j-1) + p_j(P(i-1, j-1) - P(i, j-1)), \quad (3.20)$$

$$Q(i, j) = (1-q_j)Q(i-1, j-1) + q_j Q(i, j-1) =$$
$$= Q(i-1, j-1) + q_j(Q(i, j-1) - Q(i-1, j-1)), \quad (3.21)$$

Це означає, що алгоритм Рушді вимагає 3k (n - k +1) арифметичних операцій, а його обчислювальна складність має порядок O (k (n - k +1)).

Обчислення значень $P(i, j)$ або $Q(i, j)$, показане на рис.3.1 може проводитися по рядках, по стовпцях і навіть по діагоналі. Однак, для того, щоб скоротити вимоги до пам'яті, для випадку P (3, 7), обчислення проводяться по діагоналі, з належною увагою до кордонів паралелограма. В цьому випадку для виконання алгоритму, крім пам'яті для зберігання p_i для $1 \leq i \leq n$, необхідна пам'ять для зберігання k + 1 = 4 скалярів. Вимоги до додаткової пам'яті, що виникають при вирішенні будь-якої подібної задачі, - це min {k +1, n - k} при обчисленні за стовпцями (якщо k менше) або при обчисленні по рядках (якщо n - k + 1 менше). Також, цікавою особливістю цього алгоритму є те, що і для обчислення надійності, і для обчислення ненадійності його обчислювальна складність однакова.

Результати порівняння за часом виконання і необхідної пам'яті алгоритмів Рушді і Барлоу і Хейдтманна показані в таблиці 3.2. Час виконання вимірюється кількістю операцій додавання, множення, і звернення до масиву. У таблицях 3.3 і 3.4 вибірково показані результати оцінки надійності та ненадійності 5-out-of-8: G системи відповідно, отримані за алгоритмом Рушді. В обох таблицях ми прийняли, що надійність компонент $p_j = 0{,}9 - 0{,}001(j-1)$, де $j = 1, 2, \ldots, 8$.

Варто відзначити ще одну перевагу алгоритмів Барлоу і Хейдтманна і Рушді. Всі проміжні значення, необхідні для обчислення

$P(k, n)$ або $Q(k, n)$ – це значущі числа, що представляють $P(i, j)$ або $Q(i, j)$, де $1 \le i \le k$ та $i \le j \le n - k + i$.

Таблиця 3.2 Порівняння обчислювальної і просторової складнощів

Алгоритм	Рущді (Rushdi)	Барлоу (Barlow) та Хейдтманна (Heidtmann)
Тимчасова складність		
Операції множення	$k(n - k + 1)$	$(k + 1)(n - k + 1) - 1$
Операції додавання	$2k(n - k + 1)$	$(2k + 1)(n - k + 1) - 2$
Звернення до одновимірних масивів	$4k(n - k + 1) + 1$	$(4k + 3)(n - k + 1) - 3$
Просторова складність		
Необхідна пам'ять	$\min(k + 1, n - k + 2)$	$k + 2$

Таблиця 3.3 Обчислення $P(5, 8)$ для 5-out-of-8:G системи за методом симетричної перемикальної функції

	j	0	1	2	3	4	5	6	7	8
i	0	1	1	1	1					
	1	0	0,900000	0,989000	0,998680	0,999828				
	2		0	0,801000	0,966440	0,994489	0,999081			
	3			0	0,704480	0,932437	0,985802	0,997089		
	4				0	0,613246	0,887750	0,971094	0,992930	
	5					0	0,527391	0,833697	0,949110	0,985480

Таблиця 3.4 Обчислення Q (5, 8) для 5-out-of-8: G системи за методом симетричної перемикальної функції

	j	0	1	2	3	4	5	6	7	8
	0	0	0	0	0					
	1	1	0,100000	0,011000	0,001320	0,300172				
	2		1	0,199000	0,033560	0,005511	0,000919			
i	3			1	0,295120	0,067563	0,014198	0,002911		
	4				1	0,386754	0,112250	0,028906	0,007070	
	5					1	0,472609	0,166303	0,050890	0,014520

Ці числа доступні для інженера, який розраховує надійність, без жодних додаткових витрат, і можуть допомогти йому зробити обґрунтовану економічну оцінку надмірності. Наприклад, 5-а рядок в таблиці 3.3 представляє надійність $P(5,\ j)$, де j змінюється від 5 до 8. Збільшення надійності системи, що досягається при збільшенні розміру системи від j до $j + 1$, дорівнює

$$\Delta_j P(i, j) = P(5, j + 1) - P(5, j). \tag{3.22}$$

Отже, економічний еквівалент цього збільшення надійності може бути оцінений і зрівняний з вартістю додавання додаткового компонента, і, таким чином може бути отримано оптимальну кількість компонент для 5-out-of-j системи.

Недоліком обох цих алгоритмів є те, що їх можна використовувати тільки для розрахунку надійності базових систем. А також - то що вони при великій розмірності системи стикаються з непереборними обчислювальними труднощами.

4. РОЗРАХУНОК ЙМОВІРНІСТЬ БЕЗВІДМОВНОЇ РОБОТИ ВБС, ЩО СКЛАДАЄТЬСЯ З КІЛЬКОХ ПІДСИСТЕМ

4.1 Загальний підхід до розрахунку надійності ВБС статистичним методом

GL-моделі знаходять застосування в самих різних сферах науки - в моделюванні систем зв'язку, медичної діагностики, пошуку даних в файлах, тестуванні резервованих систем і ін. Основним призначенням графо-логічних моделей, а також - і інших математичних моделей ВБС є проведення експериментів над ними .

Цілі цих експериментів можуть полягати у визначенні якісних характеристик системи, а також - у визначенні т.зв. «Вузьких місць» системи, тобто модуля або деякої сукупності модулів, відмова яких є найбільш вірогідним і призводить до виведення з ладу системи в цілому. Прикладом "вузького місця" в дубльованих системах є, як відомо, відновлює орган. У реконфігурованих ВБС з використанням різних видів надмірності подібні органи відсутні, і визначення "вузьких місць" має свої складності. особливо це стосується керівників ВБС, в яких використовуються різні процесори і різні ступені захисту від відмов навіть в рамках однієї підсистеми, і в першу чергу на етапі проектування з метою захисту від помилок розробника. Для вирішення цього завдання з моделлю ВБС може бути проведений експеримент, від повноти якого залежить ймовірність виявлення всіх "вузьких місць". Повнота експерименту може бути підвищена шляхом використання спеціалізованої комп'ютерної системи, до складу якої входить керований генератор псевдовипадкових чисел - векторів, що імітують потік відмов на вході моделі. Зауважимо, що в цьому і в наступному параграфах відмова модуля буде представлятися 1 у відповідному розряді вектора стану системи.

Отримання оцінки показників надійності з достатньою точністю і в прийнятний час для таких складних технічних об'єктів, як відмовостійкі багатопроцесорні обчислювальні системи є однією з найважчих і найважливіших проблем в сучасній техніці. Особливе значення ця проблема набуває для оптимального вибору структури програмно - апаратних засобів відмовостійких багатомодульних систем (ВБС) на етапі їх проектування. GL-модель призначена для ефективного аналізу поведінки системи, що має велике число станів. Вона дозволяє зробити істотний крок вперед у вирішенні цього завдання. Важливим в даному випадку є те, що за допомогою GL-моделі вдається досить компактно представити критерій, відповідно до якого здійснюється оцінка поведінки системи в певних умовах (наприклад, при впливі на систему потоку відмов її елементів). Однак для здійснення ефективної процедури оцінки надійності ВБС, що володіють складною структурою резервування, потрібна розробка більш загальної комплексної комп'ютерної моделі, складовою частиною якої є GL-модель. Відповідно для організації комп'ютерного моделювання поведінки системи і розрахунку показників її надійності будуть потрібні в даному випадку ряд додаткових блоків. Так само як блок генерації двійкових векторів, за допомогою яких при моделюванні представляються набори модулів системи, які опинилися в стані відмови. При цьому вважається, що інтенсивність відмов конкретних модулів відома. Залежно від обраної методики порядок генерації двійкових наборів може являти собою псевдослучайную або детерміновану послідовність, зокрема, це можуть бути послідовності наборів постійної ваги з можливістю зміни значення ваги. Фактично завдання блоку генерації полягає в моделюванні потоку відмов в залежності від поставлених попередньо вихідних даних про інтенсивності відмов окремих модулів системи і часу, для якого визначається ймовірність безвідмовної роботи системи.

Крім цього до складу моделі повинен входити блок обробки результатів, одержуваних в кожному циклі роботи блоку генерації відмов і блоку, що реалізує GL-модель. На виході даного блоку отримуємо значення розраховуються (оцінюваних) показників надійності, наприклад, ймовірності безвідмовної роботи системи за деякий заданий час, середнього часу безвідмовної роботи, значень функції готовності і т.д.

На рис. 4.1 показана узагальнена структурна схема процедури моделювання поведінки системи в потоці відмов, яка містить згадані вище блоки.

У найзагальнішому вигляді робота моделі полягає в наступному. Спочатку проводиться завантаження вихідних даних, відповідних властивостях і структурі конкретної системи, поведінка якої буде моделюватися і показники надійності оцінюватися. Далі в залежності від обраної методики проводиться настройка генератора, що імітує потік відмов. Після кожного циклу генерації формується n розрядний двійковий вектор виду $(x_1, x_2, ..., x_n)$, де значення кожного компонента вектора інтерпретується, наприклад, наступним чином: кожна змінна x_i $(i = \overline{1,n})$позначає стан i-го елемента (модуля) системи, тобто x_i=0, якщо *i*-й елемент системи працездатний, і x_i=1 в разі, якщо елемент непрацездатний (знаходиться в стані відмови.)

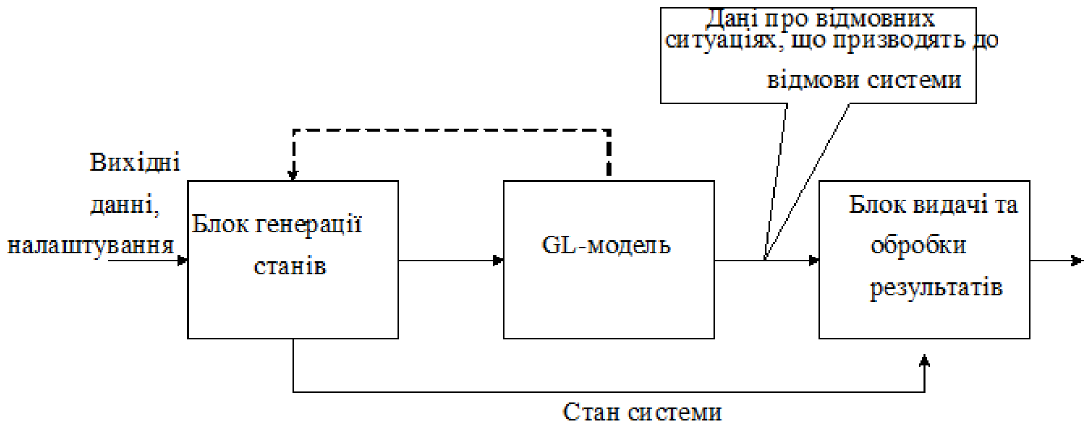


Рисунок. 4.1 - Структурна схема процедури моделювання поведінки системи в потоці відмов

Сформований двійковий вектор надходить в блок GL-моделі для виконання процедури графо-логічного моделювання і в блок обробки для розрахунку ймовірності отказовие ситуації, яку цей вектор зображує.

Завдання блоку GL-моделі полягає в тому, щоб відповісти на питання, чи приводить дана отказовие ситуація до відмови системи або ж система, використовуючи свій резерв, в змозі продовжувати роботу (маскуючи відмовили модулі або здійснюючи реконфігурацію своїх програмно - апаратних засобів). Перевага використання в загальній процедурі моделювання GL-моделювання полягає в швидкості отримання результату. Це обумовлено компактністю подання критерію, а саме у вигляді відносно нескладних булевих функцій, які відзначають ребра графа, що використовується в моделі. Обчислення значень цих функцій проводиться на наборі, сформованому в блоці генерації станів досліджуваної системи, і відбувається дуже швидко навіть при дуже великому числі модулів в системі.

Дані про отказовие ситуаціях, що призводять до відмови системи, накопичуються в блоці обробки і надалі використовуються для проведення розрахунків імовірнісних показників по тій чи іншій методиці і, природно, для визначення "вузьких місць" в ВБС.

Розглянемо трохи більш докладно методику розрахунку одного з основних показників надійності - ймовірності безвідмовної роботи або альтернативного йому показника - ймовірності відмови системи за заданий час. Якщо один з цих показників відомий, то виходячи з нього, можна отримати і інші показники, враховуючи зв'язаність їх між собою. Тут необхідно зазначити, що вибір методики підрахунку залежить чи обумовлений заданими умовами, наприклад, пов'язаними з необхідною точністю результату, допустимим часом проведення моделювання або можливостями, якими володіє використовуваний комплекс програмно-технічних засобів.

Розглянемо коротко деякі методики, відзначаючи при цьому доцільність їх використання в залежності від заданих умов.

Практично найбільш точний результат буде отримано, якщо можна здійснити за прийнятний час повний перебір всіх можливих станів (отказовие ситуацій) для досліджуваної системи. Нехай $\delta_i = (\tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_n)$ згенерований і-й набір, що відображає деяку отказовие ситуацію. Нехай також ймовірність відмови j-го елемента системи дорівнює p_j і відмови окремих елементів події незалежні. У цьому випадку ймовірність отказовие ситуації $\delta_i = (\tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_n)$ визначиться наступним чином

$$p(\delta_i) = p(\tilde{x}_1) \cdot p(\tilde{x}_2) \cdot ... \cdot p(\tilde{x}_n), \qquad (4.1)$$

де $p(\tilde{x}_i) = p_i$, якщо $x_i = 1$ та $p(\tilde{x}_i) = 1 - p_i$, якщо $x_i = 0$. З огляду на далі, що отказовие ситуації є несумісними подіями, формулу для ймовірності відмови досліджуваної системи можна записати як суму ймовірностей появи таких отказовие ситуацій, які призводять до відмови системи. Таким чином, ймовірність відмови системи Q_c визначаємо з наступного виразу

$$Q_c = \sum_{\delta_i} \alpha(\delta_i) \cdot p(\delta_i), \qquad (4.2)$$

а ймовірність безвідмовної роботи за заданий час – за формулою

$$P_c = 1 - \sum_{\delta_i} \alpha(\delta_i) \cdot p(\delta_i), \qquad (4.3)$$

де $\alpha(\delta_i)$ - коефіцієнт, що відповідає і-ой отказовие ситуації, причому, $\alpha(\delta_i) = 1$, якщо дана ситуація призводить до відмови всієї системи, і $\alpha(\delta_i) = 0$, якщо дана ситуація не призводить до відмови всієї системи.

Дану методику підрахунку показників надійності системи можна використовувати тільки для систем відносно невеликий розмірності, що і є її недоліком.

Подолати обмеження, пов'язані з необхідністю повного перебору станів, дозволяють статистичні оцінки, які виходять для вибірок досить великого обсягу, але все-таки значно менших, ніж при повному переборі. Ідея методики, заснованої на використанні статистичних даних, полягає у випадковому або псевдовипадковому генеруванні логічних змінних x_j із заданою вірогідністю p_j появи одиниці, де p_j - ймовірність відмови j-ого модуля системи. Сформований таким чином випадковий набір $\delta_i = (\tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_n)$, представляє деяку отказовие ситуацію, надходить на вхід блоку графо-логічного моделювання для визначення поведінки системи в даній ситуації. Процес обчислень повторюється N_0 раз (кожен раз з новими значеннями незалежних випадкових значень аргументів x_j). Далі підраховується кількість N (t) випадків коли процедура графо-логічного моделювання давала позитивний результат, тобто визначала, що для даних отказовие ситуацій система залишалася працездатною.

Відношення

$$P_c = \frac{N(t)}{N_0} \tag{4.4}$$

є статистичною оцінкою ймовірності безвідмовної роботи системи. Генерування випадкових логічних змінних x_j з заданою вірогідністю p_j виникнення одиниці здійснюється на підставі рівномірно розподілених в інтервалі (0, 1) випадкових величин ξ, одержуваних за допомогою стандартних програм, що входять в ПО сучасних комп'ютерів. В цьому випадку генерування значень змінних x_j с з заданою вірогідністю p_j виникнення одиниці виконується у відповідності з наступним правилом:

$x_j = 0$, якщо $\xi > p_j$ та $x_j = 1$, якщо $\xi \le p_j$

Оскільки випадкові події, що моделюються через значення x_j, повинні бути незалежними, генерування nN_0 повинні бути незалежними, генерування nN_0 - кратного генерування випадкової величини ξ. Зрозуміло, що при досить великому обсязі статистичної вибірки (а від цього залежить точність результату) може неприпустимо збільшитися час моделювання. Крім того, уповільнення моделювання відбувається через те, що відмови окремих модулів реальної системи, що моделюється є рідкісними випадковими подіями, і необхідно в значній мірі збільшувати обсяг вибірки для отримання достатньої кількості ненульових наборів. Остання обставина певною мірою можна обійти за рахунок введення нормувального коефіцієнта. Цей коефіцієнт повинен бути підібраний таким чином, щоб, сума значень ймовірностей появи отказовие ситуацій, попередньо помножених на цей коефіцієнт, дорівнювала одиниці. У цьому випадку поява будь-якої з отказовие ситуацій представляє достовірна подія. Іншими словами, в кожному випробуванні з'являється якась отказовие ситуація. В результаті можна зменшити необхідний обсяг вибірки і загальний час моделювання. Остаточний результат, який обчислюється за формулою (4.4), необхідно розділити на нормувальний коефіцієнт.

Необхідно відзначити, що використання статистичного експерименту дає лише наближене значення величини Pc, у вигляді статистичної оцінки. Ефективне значення похибка оцінки, одержаній відповідно до вираження (4.4), визначається за формулою Муавра - Лапласа і дорівнює

$$\sigma \approx \sqrt{P(1-P) / N_0}\,, \tag{4.5}$$

де P значення P_{c} для довільного моменту часу. Довірчий інтервал не повинен перевищувати наступних меж

$$P \pm 2\sqrt{P(1-P) / N_0}\,. \tag{4.6}$$

З цих виразів з очевидністю випливає, що для забезпечення малої похибки потрібна велика кількість випробувань, в результаті чого зростає час моделювання і розрахунку.

Перейдемо тепер до розгляду таких методик, які дозволяють отримати оцінку з заданою точністю за прийнятний час моделювання.

Точність розрахунку також як і точність будь-яких вимірювань - це ступінь збігу результатів розрахунків (вимірювань) з ідеальним значенням розраховується вимірюваної величини. Кількісно точність розрахунку визначається помилкою розрахунку, яка виходить як різниця між фактичним (отриманим в результаті моделювання) і ідеально точним значенням результату.

Збільшення точності пов'язане практично завжди зі збільшенням машинного часу і подорожчанням відповідної роботи. Тому слід прагнути не просто до високої точності і достовірності, а до доцільною точності і достовірності. У кожному конкретному випадку можуть бути свої ознаки доцільною точності і достовірності моделювання. У нашому випадку при розрахунку надійності ВБС істотним є те, що інтенсивності відмов окремих модулів системи дуже малі і відповідно дуже малі ймовірності відмов модулів за відносно невеликий час, для якого проводиться розрахунок. Відповідно ймовірності отказовие ситуацій, одержуваних за допомогою виразу (4.1) також вельми маленькі числа. Зі збільшенням кратності відмов ці числа різко зменшуються. Тому цілком реально припустити, що, починаючи з певної кратності, внеском залишилися отказовие ситуацій, що призводять до відмов системи, можна знехтувати в силу його мізерно малих значень. Природним критерієм тут виступає задана (допустима) точність розрахунку.

Нехай точність оцінки, яка повинна бути витримана, визначається величиною ΔP. Точність розрахунку позначимо ΔP_0. Тоді моделювання і розрахунок слід проводити до тих пір, поки виконується нерівність $\Delta P_0 > \Delta P$. Нехай також k - це величина кратності, до якої можливий перебір всіх станів системи і розрахунок проводиться відповідно до виразами (4.2) і (4.3). Далі запишемо вираз, за допомогою якого можна визначити величину похибки для деякої кратності відмов. Необхідно відзначити, що при великій розмірності системи перебрати все отказовие ситуації або вектора $\delta_i = (\tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_n)$, починаючи з деякої величини кратності відмов k, виявляється практично неможливим. Тому зробити точний розрахунок величини ΔP_0, наприклад, за формулою (4.2) також неможливо. У зв'язку з цим необхідно ввести деякі припущення, що спрощують розрахунок величини ΔP_0. Ці припущення необходимо ввести таким чином, щоб оцінка величини ΔP_0 була, по крайней мере, не менше реального значення цієї величини. Одне з таких припущень полягає в тому, що передбачається, що відмови всіх модулів різновірогідні і рівні максимальної ймовірності відмови для модулів системи, (ймовірності відмови самого ненадійного модуля.),тобто

$p(x_j) = p_m$. Наступне припущення полягає в тому, що передбачається, що досліджувана система має властивість монотонності в сенсі її реакції на отказовие ситуації. Під цією властивістю будемо розуміти наступне. Нехай система відмовляє при появі деякої отказовие ситуації, що подається набором $\delta_i = (...,\alpha_q,...,\alpha_r,...,\alpha_s,...)$ з вагою k і конфігурацією одиничних компонентів на позиціях $(...,\alpha_q,...,\alpha_r,...,\alpha_s,...)$.Дана система вважається монотонної, якщо вона завжди відмовляє також при будь-якій іншій отказовие ситуації, представленої набором $\delta_u = (...,\alpha_q,...,\alpha_r,...,\alpha_s,...)$, що має вагу більшу, ніж k, але зберігає поодинокі компоненти на позиціях $(...\alpha_q,...\alpha_r,...,\alpha_s...)$.Легко помітити, що для монотонних систем мінімальне число відмов системи для відмовних ситуацій з кратністю k + 1 дорівнює числу відмов системи N (k) для відмовних ситуацій з кратністю k. У той же час максимальне число відмов для відмовних ситуацій з кратністю k + 1 дорівнює $N(k+1)_{\max} = C_n^{k+1}$. Якщо вважати, що середнє число відмов системи R (k + 1) для отказовие ситуацій з кратністю k + 1 визначиться відповідно до вираження

$$R(k+1) = \frac{N(k) + N(k+1)_{\max}}{2}, \quad (4.7)$$

то величини R (k) для різних значень кратності k обчислюються наступним чином

$$R(1) = \frac{N_k + C_n^{k+1}}{2}; \qquad (4.8)$$

$$R(i) = \frac{R(i-1) + C_n^{k+i}}{2} \text{ для } i > 1. \qquad (4.9)$$

З огляду на вищевикладене, шукану похибка розрахунку визначаємо за допомогою формули

$$\Delta P_0 = \sum_{i=1}^{n-k} \frac{R(i) + C_n^{k+i}}{2} \cdot p^{k+i} \cdot (1-p)^{n+k-i} \qquad (4.10)$$

Однак в деяких випадках може виявитися, що продуктивності готівки (доступних) технічних засобів все ж недостатньо для виконання моделювання та розрахунку за прийнятний час за умови досягнення необхідної точності. У цьому випадку доцільно використовувати останню розглянуту методику в поєднанні з методом статистичного моделювання, про який йшлося вище. Орієнтовна схема розрахунку в даному випадку може виглядати наступним чином. Спочатку виконується повний перебір станів системи (отказовие ситуацій) і для них виконуються необхідні процедури моделювання для величин кратності, які дозволяють виконати цю роботу за прийнятний час. Якщо при цьому виявляється, що величина розрахункової похибки, яка підраховується для даної кратності k, наприклад, за формулою (4.10), більше допустимої, тобто має місце $\Delta P_0 > \Delta P$, то можна збільшити кратність відмов при моделюванні. Однак для того, щоб піти від неприйнятного обсягу перебору, який виникає при цьому, далі, при розгляді великих, ніж величина k кратності, використовується методика статистичного експерименту зі значно меншим числом розглянутих станів системи. Безумовно, при цьому виникає додаткова похибка, яку також необхідно враховувати. З огляду на, що ця додаткова похибка залежить від числа реалізацій при статистичних випробуваннях відповідно до формулами (4.5) і (4.6), можна оптимальним чином підібрати величини кратності k, при якій можливий повний перебір, величини кратності, в межах яких здійснюється статистична вибірка і, нарешті, число статистичних випробувань N_0 (обсяг статистичної вибірки). Передбачається, що при такому оптимальному виборі перерахованих вище параметрів повинні бути задоволені як вимоги по точності результату, так і не перевищено допустимий час для обох задач: пошук "вузьких місць" і розрахунок надійності.

Безпосередньо процедура статистичного моделювання відмовних ситуацій може бути організована при реалізації кожного випробування в два етапи. На першому етапі здійснюється равновероятности вибір однієї з величин кратності відмов в межах допустимого діапазону (k1, k2), який встановлюється попередньо відповідно до вищевикладених міркуваннями. По суті справи проводиться равновероятности генерація одного з цілих чисел із зазначеного діапазону. Реалізація зазначеної процедури досить проста і може бути виконана з великою швидкістю з допомогою програмних або апаратних засобів. На другому етапі проводиться настройка генератора псевдовипадкових рівноважних наборів на формування чергового набору з вагою, рівним псевдовипадковому цілому числу, отриманому на першому етапі. Далі здійснюється формування многоразрядного набору із заданою вагою.

Розглянемо, яким чином в даному випадку можна здійснити розрахунок відповідної частки ймовірності безвідмовної роботи досліджуваної системи. Йдеться про безвідмовної роботи системи, що складається з n модулів при проведенні статистичного експерименту (моделюванні) для додаткового діапазону кратності відмов. Нехай розглядається діапазон кратності визначено в межах від k_r до k_q. Нехай також в результаті статистичного експерименту для кратності k_j ($k_r \le k_j \le k_q$) отримано, що при проведенні $N_0^{(j)}$ статистичних випробувань число випадків, коли система залишалася працездатною дорівнює величині $N^{(j)}(t)$. Крім цього для простоти будемо вважати, що ймовірності відмов всіх модулів системи однакові і рівні ймовірності відмови найменш надійного модуля в системі. Останнє припущення не повинно викликати заперечень, якщо згадати, що обсяг статистичної вибірки встановлюється в залежності від допустимої похибки розрахунку, яка в свою чергу визначається при таких же припущеннях.

З огляду на сказане, можемо величину ймовірності появи відмови системи в разі впливу відмов модулів кратності k_j визначити наступним чином

$$p_c^{(j)} = \left(1 - \frac{N^{(j)}(t)}{N_0^{(j)}}\right) \cdot C_n^{k_j} \cdot p_m^{k_j} \cdot (1 - p_m)^{n-k_j} . \quad (4.11)$$

Величина ймовірності відмови системи для випадку появи отказовие ситуацій для всього розглянутого діапазону (k_r, k_q) буде дорівнювати

$$p_c^{(r,q)} = \sum_{r \le j \le q} \left(1 - \frac{N^{(j)}(t)}{N_0^{(j)}}\right) \cdot C_n^{k_j} \cdot p_m^{k_j} \cdot (1 - p_m)^{n-k_j} . \quad (4.12)$$

При здійсненні першого етапу можливий і інший варіант вибору поточної кратності, а саме з урахуванням певних терезів для кожної величини кратності, що реалізується у вигляді різної ймовірності появи чисел, які обирають з діапазону (k_r, k_q). На користь такого варіанту існує ряд міркувань, однак, ми на цьому тут зупинятися не будемо.

При виборі кратності модельованих відмов для прискорення моделювання, доцільно враховувати різний рівень відмовостійкості, який притаманний різним підсистемам досліджуваної ВБС. Це означає, що при формуванні в процесі моделювання багаторозрядних векторів, що відображають відмовні стани системи, потрібно виділяти підмножини розрядів, відповідних модулів певної підсистеми. Для набору, складеного з такого виділеного підмножини розрядів, вага задається не менший, ніж рівень відмовостійкості відповідної підсистеми.

4.2 Розрахунок надійності ВБС при розбитті вихідної системи на підсистеми

Розглянутий вище метод дозволяє розраховувати надійність для ВБС з великою кількістю елементом, однак, при його використанні доводиться оперувати з векторами стану системи відповідної довжини. Чим більше кількість елементів системи, тим довше її вектор стану і тим більше часу і ресурсів буде займати розрахунок надійності. А також - збільшуватися похибка обчислень, оскільки, навіть використовуючи метод статистичних оцінок, зі збільшенням довжини вектора стану системи то кількість їх, на якому можна в прийнятний час промоделювати реакцію системи буде скорочуватися.

Можна скоротити час розрахунку надійності системи з великою кількістю елементів, якщо цю систему розбити на кілька підсистем меншою розмірності і потім, виходячи з отриманих результатів, обчислити надійність системи в цілому.

Нехай є якась ВБС (не обов'язково базова), що складається з

n елементів, яку можна розбити на l багатопроцесорних підсистем, що складаються з n_1, n_2, …, n_l елементів і стійких до відмови m_1, m_2, …, m_l елементів відповідно. Нехай ці підсистеми не містять загальних один з одним елементів (для спрощення ситуації). Відмови елементів - незалежні одна від одної події, а оскільки підсистеми не містять спільних елементів, то і відмови підсистем події незалежні.

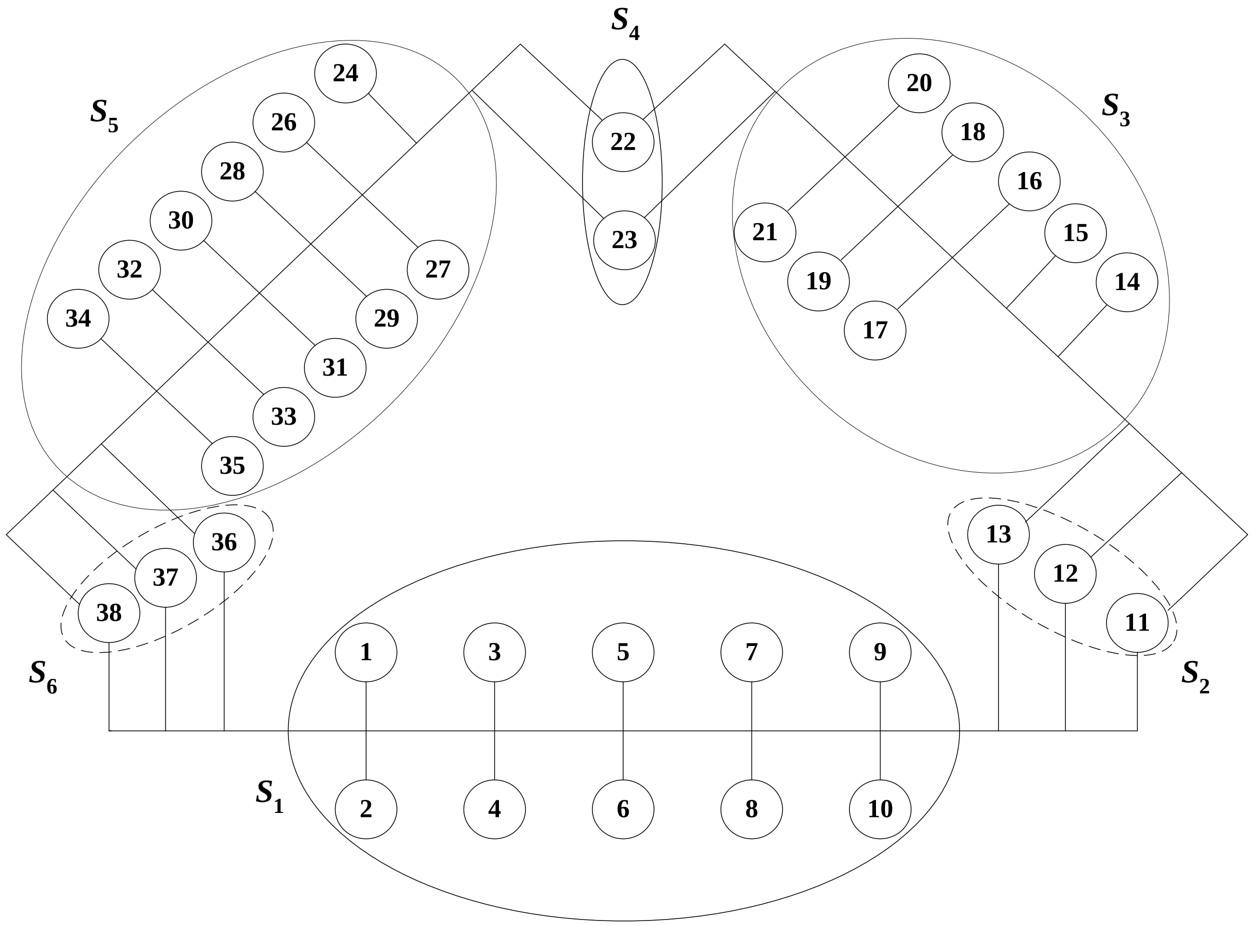


рис. 4.2

Необхідно розрахувати надійність кожної з цих підсистем методом статистичного моделювання, що базується на експерименті з

GL-моделями виділених в системі підсистем, і на підставі отриманих даних обчислити надійність системи в цілому.

Для того, щоб спростити запис формул, будемо вважати, що ймовірності відмов елементів системи $p_1, p_2, \ldots, p_n$ рівні між собою

$$p_1 = p_2 = p_3 = \ldots = p_n = p$$

і, відповідно, рівні їх ймовірності їх безвідмовної роботи

$$q_1 = q_2 = q_3 = \ldots = q_n = q = 1 - p$$

Тоді ймовірність появи певного вектора системи v_i дорівнює

$$p_{v_i} = p^i q^{n-i}, \tag{4.13}$$

де i – число відмовивших елементів.

Спочатку ми розраховуємо ймовірність безвідмовної роботи не система в цілому, а ймовірності безвідмовної роботи підсистем. Отже, для генерації векторів стану підсистем нам необхідно знати ймовірності їх появи.

Імовірність появи вектора стану підсистеми, якщо не враховувати той факт, що цей вектор може з'являтися в певній кількості векторів стану системи, як підвектора дорівнює

$$p_{v_{i_j}} = p^{i_j} q^{n_j - i_j}, \tag{4.14}$$

де n_j та i_j – кількість елементів j-ої підсистеми і кількість відмовили в цій підсистемі елементів відповідно ($j = \overline{1, l}$).

Кількість же векторів стану системи, в яких може з'явитися даний підвектора, так само 2^{n-n_j} штук. Ці вектора, крім заданого підвектора, містять в собі по k_i нулів і по $n - n_j - k_j$ одиниць (k_j=0,….,n–n_j–k_j). Причому кількість векторів, що містять рівно по k_{i} нулів і по $n - n_j - k_j$ одиниць, дорівнює

$$C(n - n_j - k_j,\, k_j) = \frac{(n - n_j)!}{(n - n_j - k_j)!\,k_j!}. \tag{4.15}$$

Імовірність появи вектора, що містить даний підвектора, дорівнює

$$p_{v_i} = C(n-n_j-k_j,\, k_j)p^{k_j}q^{n-n_j-k_j}p^{m_j}q^{n_j-m_j} = \\ = C(n-n_j-k_j,\, k_j)p^{m_j+k_j}q^{n-k_j-m_j} \tag{4.16}$$

Тоді ймовірність появи конкретного вектора стану підсистеми буде дорівнює:

$$p_{v_{i_j}} = (\sum_{j=0}^{n-n_j} C(n-n_j-k_j,\, k_j)p^{k_j}q^{n-n_j-k_j})p^{m_j}q^{n_j-m_j} \tag{4.17}$$

Але не всі з них треба розглядати, так як серед них є вектори, які можуть не привести до відмови даної підсистеми, але приведуть до відмови системи в цілому.

Далі надійність підсистеми можна розрахувати по одному з описаних в параграфі 4.1 методів. Метод необхідно вибирати виходячи з розмірності системи і необхідної точності розрахунку. Скорегуємо формули, наведені в параграфі 4.1, для розрахунку надійності підсистем.

Якщо розмірність підсистем така, що за прийнятний час можна здійснити повний перебір всіх можливих відмовних ситуацій, то тоді ймовірність безвідмовної роботи кожної конкретної підсистеми можна обчислити за такою формулою, отриманої виходячи з (4.2) і (4.17):

$$Q_c = \sum_{v_{i_j}} \alpha(v_{i_j}) \cdot p_{v_{i_j}} = \\ = (\sum_{j=0}^{n-n_j} C(n-n_j-k_j,\, k_j)p^{k_j}q^{n-n_j-k_j})) \cdot (\sum_{v_{i_j}} \alpha(v_{i_j})p^{m_j}q^{n_j-m_j}), \tag{4.18}$$

а ймовірність безвідмовної роботи за заданий час - за формулою

$$P_c = 1 - \sum_{v_{i_j}} \alpha(v_{i_j}) \cdot p_{v_{i_j}} = \\ = 1 - (\sum_{j=0}^{n-n_j} C(n-n_j-k_j,\, k_j)p^{k_j}q^{n-n_j-k_j})) \cdot (\sum_{v_{i_j}} \alpha(v_{i_j})p^{m_j}q^{n_j-m_j}), \tag{4.19}$$

де $\alpha(v_{i_j})$ – коефіцієнт, що відповідає j–ої відмовній ситуації для i-ої підсистеми, причому, $\alpha(v_{i_j})$ = 1, якщо дана ситуація призводить до відмови всієї системи, і $\alpha(v_{i_j})$ = 0, якщо дана ситуація не призводить до відмови всієї системи. А $p_{v_{i_j}}$ - ймовірність появи *j*–ої відмовної ситуації для *i*-ої підсистеми. Все інше аналогічно як для системи в цілому.

Якщо розмірність підсистем не дозволяє здійснити повний перебір відмовних ситуацій, то тоді їх надійність можна обчислити за допомогою методу статистичного розрахунку показників надійності аналогічно, як і для систем. Точно таким же методом формуються випадкові набори $v_{i_j} = (\tilde{x}_{i_1}, \tilde{x}_{i_2}, ..., \tilde{x}_{i_{nl}})$, що представляють отказовие ситуацію, з тією лише різницею, що генерування випадкових змінних x_{i_j} здійснюється не з ймовірністю р появи 1, а з ймовірністю

$$p_{i_j} = p\sum_{j=0}^{n-n_j} C(n-n_j-k_j,\, k_j)p^{k_j}q^{n-n_j-k_j}, \qquad (4.20)$$

Статистична оцінка ймовірності безвідмовної роботи підсистеми обчислюється відповідно до формули (4.4).

Аналогічно, при досить великому обсязі статистичної вибірки, вводяться нормуючі коефіцієнти, які підбираються точно таким же чином, як і для розрахунку надійності системи в цілому. Остаточний результат, який обчислюється за формулою (4.4), необхідно розділити на підібраний нормувальний коефіцієнт.

Поєднати необхідну точність розрахунку і прийнятний час моделювання, як і в разі розрахунку надійності системи в цілому можна, якщо враховувати той факт, що інтенсивності відмов окремих модулів підсистеми дуже малі і відповідно дуже малі ймовірності відмов модулів за відносно невеликий час, для якого проводиться розрахунок. Відповідно ймовірності отказовие ситуацій, одержуваних за допомогою виразу (4.17) також вельми маленькі числа. А зі збільшенням кратності відмов ці числа різко зменшуються. Отже, як і в разі розрахунку надійності системи в цілому, можна припустити те що, починаючи з певної кратності внеском залишилися отказовие ситуацій можна знехтувати. І керуватися тут також треба допустимою точністю розрахунку для даної підсистеми. Різниця між розрахунком надійності для підсистеми і для системи в цілому, згідно з цією методикою, полягає лише в тому, що розрахунок проводиться не за формулами (4.2) і (4.3), а за формулами (4.18) і (4.19) і похибка обчислюється не за формулою (4.10), а за такою формулою:

$$\Delta P_0 = \sum_{i=1}^{n-k} \frac{R(i) + C_n^{k+i}}{2} \cdot \left(\sum_{j=0}^{n-n_j} C(n - n_j - k_j,\, k_j) p^{k_j} q^{n-n_j-k_j} \right) \cdot p^{k+i} q^{n_j-k-i} \tag{4.21}$$

При розрахунку надійності підсистем може виявитися, що продуктивності доступних обчислювальних засобів недостатньо для виконання моделювання та розрахунку за прийнятний час за умови досягнення необхідної точності. Тоді, аналогічно як для розрахунку надійності систем в цілому, доцільно використовувати останню розглянуту методику в поєднанні з методом статистичного моделювання, про який йшлося вище. Тут також все аналогічно розрахунку надійності системи в цілому, крім формул, по якій відбувається обчислення величини ймовірності появи відмови системи в разі впливу відмов модулів кратності m_j і величини ймовірності відмови системи для випадку появи отказовие ситуацій для всього розглянутого діапазону (m_r, m_q). Вони будуть виглядати наступним чином, відповідно:

$$p_c^{(j)} = \left(1 - \frac{N^{(j)}(t)}{N_0^{(j)}} \right) \cdot C_{n_j}^{m_j} \cdot \left(\sum_{j=0}^{n-n_j} C(n - n_j - k_j,\, k_j) p^{k_j} q^{n-n_j-k_j} \right) \cdot p^{m_j} q^{n_j-m_j} . \tag{4.22}$$

$$p_c^{(r,q)} = \sum_{r \le j \le q} \left(1 - \frac{N^{(j)}(t)}{N_0^{(j)}} \right) \cdot C_{n_j}^{m_j} \cdot \left(\sum_{j=0}^{n-n_j} C(n - n_j - k_j,\, k_j) p^{k_j} q^{n-n_j-k_j} \right) \cdot p^{m_j} q^{n_j-m_j} \tag{4.23}$$

Для подальшого розрахунку надійності системи в цілому приймемо виділені раніше підсистеми як неподільні елементи системи з можливостями безвідмовної роботи рівними ймовірностями безвідмовної роботи цих підсистем. Далі необхідно побудувати модель для отриманої системи з елементами-підсистемами і розрахувати її надійність одним з розглянутих вище методів.

ВИСНОВКИ

У роботі на отримання кваліфікаційно-освітнього рівня магістра запропонована методика розрахунку надійності відмовостійких багатопроцесорних систем, що базується на використанні статистичного експерименту з GL-моделями з використанням розбиття системи на підсистеми.

У даній роботі були розглянуті найбільш ефективні методи розрахунку надійності відмовостійких багатопроцесорних систем, зокрема - метод мінімальних шляхів і розрізів, метод Барлоу і Хейдтманна і метод Рущді для базових k-out-of-n систем. А також - методика розрахунку надійності системи, заснована на статистичному експерименті з графо-логічними моделями ВБС, яка може бути застосована як для базових, так і для небазових ВБС.

Запропонована методика оцінки надійності ВБС за своїми основними принципами аналогічна методика розрахунку надійності системи, заснована на статистичному експерименті з графо-логічними моделями. Їх відмінність полягає в тому, що в запропонованій методиці система розбивається на підсистеми з меншою розмірністю і розраховується спочатку їх надійність, а потім, на основі отриманих результатів - надійність системи в цілому. Застосування цієї методики дозволяє розраховувати надійність систем з великою кількістю елементів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Каравай М.Д., Согомонян Е.С., Шагаев И.В. Відмовостійкі обчислювальні системи // Підсумки науки і техніки. Сер. Технічна кібернетика.- 1989.- Т. 28.- С. 77-118.
2. Романкевич А.М., Карачун Л.Ф., Романкевич В.А. Графо-логічні моделі для аналізу складних відмовостійких обчислювальних систем // Електронне моделювання.–2001.–т.23, №1.– с. 102-111.
3. Романкевич В.А. Про одну модель поведінки відмовостійкої багатопроцесорної системи // Радіоелектроніка та інформатіка.-1999.- №1.- С.75-76.
4. Романкевич О.М., Карачун Л.Ф., Романкевич В.О. До питання побудова моделі поведінкі багатомодульніх систем // Наукові вісті НТУУ "КПІ". - 1998. - №1. - с. 38-40.
5. А.М. Романкевич, В.В. Іванов, В.А Романкевич. Аналіз відмов багатомодульних систем зі складним розподілом відмов на основі циклічних GL-моделей // Електронне моделірованіе.-№5, т.26, 2004, с.67-81.
6. Романкевич О.М, Романкевич В.О., Кононова А.А., Рабах Ал Шбул Про деякі особливості GL-моделей K (2, n) // Вісник НТУУ "КПІ" .- Інформатика, управління та ОТ.- 2004. №41.-С.85-92.
7. Романкевич О.М., Романкевич В.О., Богуславський О.В., Ал Шбул Рабах. Аналіз відмовостійкіх багатопроцесорніх систем на основе графо-логічніх моделей нециклічного типу // Вісник технологічного університету Поділля. - 2002. - Т.2, №3. - с. 30-33.
8. Романкевич В.А., Потапова Є.Р., Ал Шбул Рабах Об'єднання моделей підсистем в рамках GL-моделей // Інформаційно-керуючі системи на залізнічному транспорті. - 2002.- №4,5.- С. 22.
9. Додонов О.Г., Кузнєцова М.Г., Горбачек Е.С. Введення в теорію живучості обчислювальних систем.- К .: Наукова думка, 1990.- 182с

10. Березюк Н.Т., Гапунін А.Я., Підлісний Н.І. Живучість мікропроцесорних систем управління.- К .: Техніка, 1989.- 144 с.
11. Lee Y. H., Shin K. G. Design and evaluation of a fault-tolerant multiprocessor, using hardware recovery block // IEEE Trans. Comput.- 1984.- Vol.C-33, №2.- P.113-124.
12. Anderson T., Lee P.A. Fault Tolerance Terminology Proposals // IEEE.- 1982.-№2.- P.29-32.
13. Прикладна теорія цифрових автоматів / Самофалов К.Г.,
14. Романкевич А.М., Валуйський В.М., Канівський Ю.С., Піневича М.М. К .: Вища школа, 1987, 375с.
15. Salam Salloum, Melvin A. Breuer. Fast Optimal Diagnosis Procedures for k-out-of-n: G Systems // IEEE Trans. on Reliability.- 1997.-Vol.46, №2.- P.283-290.
16. Chen R.W., Hwang F.K., Li Wen-Ching. Consecutive 2-out-of-n: F systems with node and link failures // IEEE Trans. on Reliability.- 1993.-Vol.42, №3.- P.497-502.
17. Іиуду К.А. Надійність контроль і діагностика обчислювальних машин і систем // М .: Вища школа - 1989, с.4 - 39.
18. Бельдхарі Н., Болдак А.А., Чебаненко Т.М. Забезпечення надійності комп'ютерних систем. // ВЕК + Київ - 1998. - с.8 - 35
19. Пацюра І.В., Корнійчук В.І., Довбиш Л.В. Надійність електронних систем // СВІТ Київ - 1997, с.2 - 20.
20. Румшіскій Л.З. Елементи теорії ймовірностей // М .: Наука - 1976,
21. 240с.
22. Крайник А.В., Курдіков Б.А., Лебедєв О.М., Недосекин Д.Д.,
23. Подобєд М.В., Полянська Т.І., Чернявський Е.А. Імовірнісні методи в обчислювальній техніці // М .: Вища школа - 1986, 312с.
24. Е.С. Вентцель Теорія ймовірностей // М .: Вища Школа - 2002 575с.
25. Кульбак С. Теорія інформації та статистика // М .: Наука - 1967 408с.

26. Way Kuo, Ming J. Zuo Optimal Reliability Modeling. Principles and Applications // John Wiley & Sons, Inc. - 2002 p.15 - 62, 85 - 140, 148 - 164, 231 - 281.
27. А. Кофман. Введення в прикладну комбінаторіку.- М .: Наука, 1975.- 479с.
28. Романкевич В.А., Морозов К.В., Фесенюк А.П. Об одном методе модификации рёберных функций GL-моделей // Радіоелектронні і комп‘ютерні системи.-№6, 2014.- С.95-99

29. Романкевич В.А., Малышева М.О., Примак И.К. Об одном способе формирования рёберных функций GL-модели // Системный анализ и информационные технологии: материалы 18-й Международной научно-практической конференции SAIT 2016.- К.: УНК «ИПСА» НТУУ «КПИ», 2016.- С.406

30. Морозов К.В., Романкевич А.М., Романкевич В.А. О характере влияния модификации рёберных функций GL-модели на её поведение в потоке отказов // Радіоелектронні і комп‘ютерні системи.-№6, 2016.- С.108-112

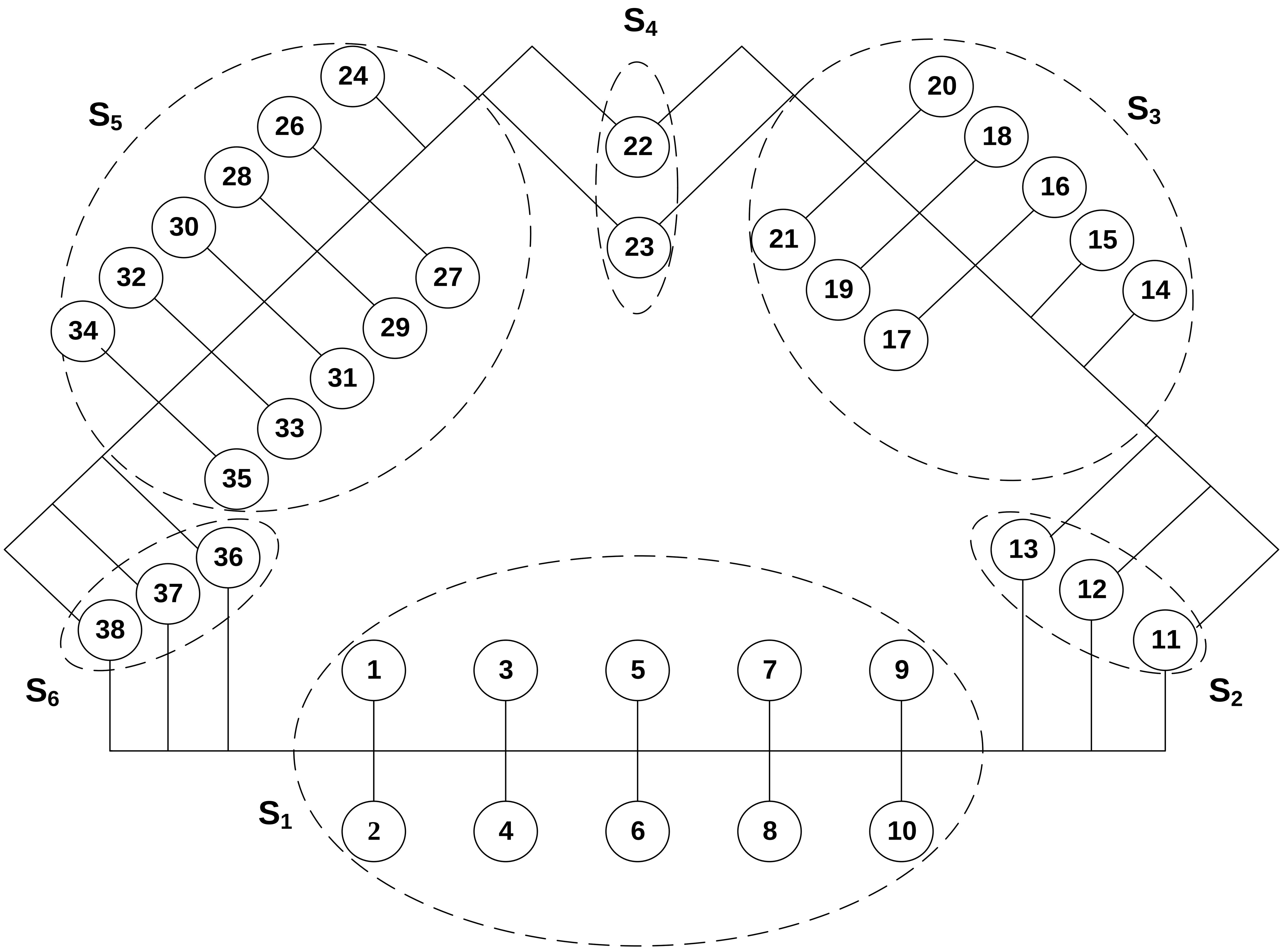
31. Морозов К.В., Романкевич В.А., Сапсай Т.Г. О модификации графо-логической модели для систем и их компонентов с множественными состояниями // Системный анализ и информационные технологии: материалы 19-й Международной научно-практической конференции SAIT 2017.- К.: УНК «ИПСА» КПИ им. Игоря Сикорского, 2017.- С.94

32. Романкевич В.О. Методи і засоби оцінки технічних характеристик гарантоздатності відмовостійких багатопроцесорних систем управління складними об’єктами: Дис. на здобуття наукового ступеня доктора технічних наук: 05.13.05; - Захищена 29.01.2018; Затв. 20.03.2018.- К., 2017.- 388с.

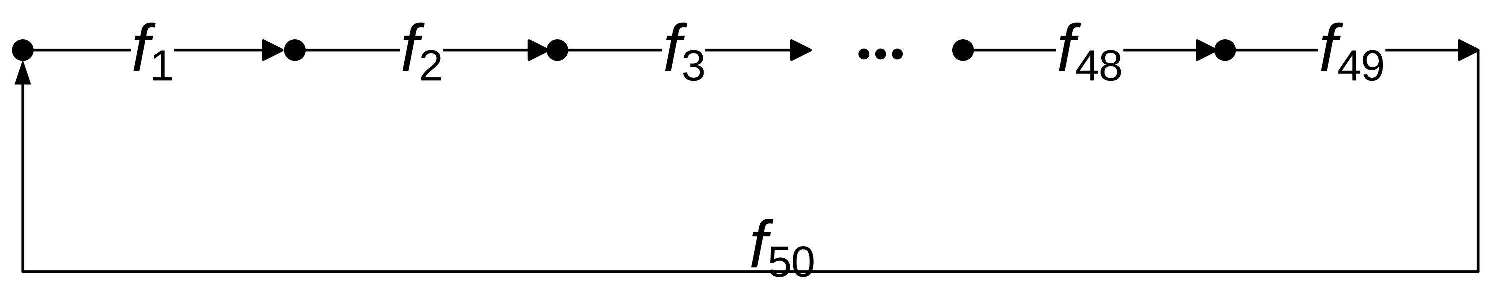
ДОДАТОК 1. КОПІЇ ГРАФІЧНИХ МАТЕРІАЛІВ

ДОДАТОК 2. ДОКУМЕНТ ПРО ВПРОВАДЖЕННЯ

ПОБУДОВА GL-МОДЕЛЕЙ ДЛЯ ПІДСИСТЕМ ВІДМОВОСТІЙКОЇ БАГАТОМОДУЛЬНОЇ СИСТЕМИ



Графо-логічна модель базової К (5, 10) підсистеми S1 циклічного типу, побудована неоптимізованими способом

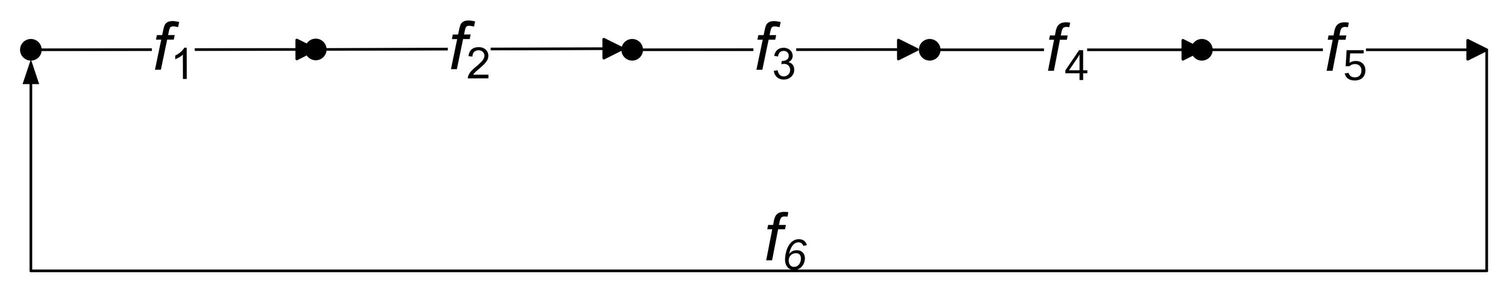


$$f_1 = x_1x_2 \vee x_3x_4 \vee x_5x_6 \vee x_7x_8 \vee x_9x_{10}$$
$$f_2 = x_1 \vee x_2 \vee x_5x_6 \vee x_7x_8 \vee x_9x_{10}$$
$$f_3 = x_1 \vee x_2 \vee x_5x_6 \vee x_7x_8 \vee x_9x_{10}$$

...

$$f_{49} = x_3x_4 \vee x_7 \vee x_8 \vee x_9 \vee x_{10}$$
$$f_{50} = x_1x_2 \vee x_7 \vee x_8 \vee x_9 \vee x_{10}$$

Графо-логічна модель базової К (5, 10) підсистеми S1 циклічного типу, побудована оптимізованим способом



$$f_1 = x_1 \vee x_2 \vee x_3 \vee x_4 \vee x_5$$
$$f_2 = (x_1x_2 \vee x_3 \vee x_4 \vee x_5)(x_1 \vee x_2 \vee x_3x_4 \vee x_5)\cdot$$
$$\cdot(x_1 \vee x_2 \vee x_3 \vee x_4) \vee x_6x_7x_8x_9x_{10}$$
$$f_3 = (x_1x_2 \vee x_3x_4 \vee x_5)(x_1 \vee x_2 \vee x_3x_4)\cdot$$
$$\cdot(x_1x_2 \vee x_3 \vee x_4)(x_1 \vee x_2 \vee x_5)(x_3 \vee x_4 \vee x_5) \vee$$
$$\vee(x_6x_7 \vee x_8x_9)(x_6x_7 \vee x_{10})(x_8x_9 \vee x_{10})\cdot$$
$$\cdot(x_6 \vee x_7)(x_8 \vee x_9)$$
$$f_4 = (x_6x_7 \vee x_8x_9 \vee x_{10})(x_6 \vee x_7 \vee x_8 \vee x_9)\cdot$$
$$\cdot(x_6x_7 \vee x_8x_9)(x_6 \vee x_7 \vee x_{10})(x_8 \vee x_9 \vee x_{10}) \vee$$
$$\vee(x_1x_2 \vee x_3x_4)(x_1x_2 \vee x_5)(x_3x_4 \vee x_5)\cdot$$
$$\cdot(x_1 \vee x_2)(x_3 \vee x_4)$$
$$f_5 = x_1x_2x_3x_4x_5 \vee (x_6x_7 \vee x_8x_9 \vee x_{10})\cdot$$
$$\cdot(x_6 \vee x_7 \vee x_8x_9 \vee x_{10})(x_6 \vee x_7 \vee x_8 \vee x_9)$$
$$f_6 = x_6 \vee x_7 \vee x_8 \vee x_9 \vee x_{10}$$

Корнєйчик Б.А. КВ-63М

ФОРМУЛИ ДЛЯ РОЗРАХУНКУ НАДІЙНОСТІ ВБС СТАТИСТИЧНИМ МЕТОДОМ З РОЗБИВКОЮ ВИХІДНОЇ СИСТЕМИ НА ПІДСИСТЕМИ

- Імовірність появи вектора снану *j* підсистеми, що складається із n_j *елементів*

$$p_{v_{i_j}} = \left(\sum_{j=0}^{n-n_j} C(n-n_j-k_j, k_j)\, p^{k_j} q^{n-n_j-k_j} \right) \cdot p^{m_j} q^{n_j-m_j}$$

- Ненадійність і надійність j-ой підсистеми, яка обчислюється при повному переборі векторів стану підсистеми

$$Q_c = \sum_{v_{i_j}} \alpha(v_{i_j}) \cdot p_{v_{i_j}} = \left(\sum_{j=0}^{n-n_j} C(n-n_j-k_j, k_j) p^{k_j} q^{n-n_j-k_j} \right) \cdot \left(\sum_{v_{i_j}} \alpha(v_{i_j}) p^{m_j} q^{n_j-m_j} \right)$$

$$P_c = 1 - \sum_{v_{i_j}} \alpha(v_{i_j}) \cdot p_{v_{i_j}} = 1 - \left(\sum_{j=0}^{n-n_j} C(n-n_j-k_j, k_j) p^{k_j} q^{n-n_j-k_j} \right) \cdot \left(\sum_{v_{i_j}} \alpha(v_{i_j}) p^{m_j} q^{n_j-m_j} \right)$$

- Похибка розрахунку надійності j підсистеми, при заданій точності оцінки, яка повинна бути витримана

$$\Delta P_0 = \sum_{i=1}^{n-k} \frac{R(i) + C_n^{k+1}}{2} \cdot \left(\sum_{j=0}^{n-n_j} C(n-n_j-k_j, k_j)\, p^{k_j} q^{n-n_j-k_j} \right) \cdot p^{k+i} q^{n_j-k-i}$$

- Імовірність появи відмови підсистеми в разі впливу відмов модулів кратності m_j

$$p_c^{(j)} = \left(1 - \frac{N^{(j)}(t)}{N_0^{(j)}} \right) \cdot C_{n_j}^{m_j} \cdot \left(\sum_{j=0}^{n-n_j} C(n-n_j-k_j, k_j)\, p^{k_j} q^{n-n_j-k_j} \right) \cdot p^{m_j} q^{n_j-m_j}$$

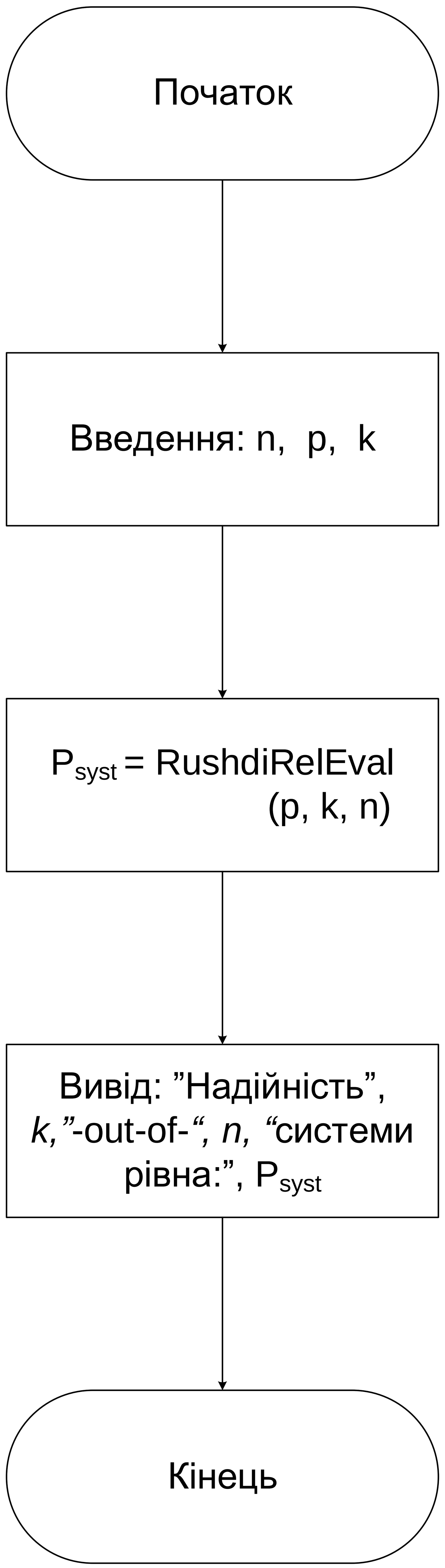
- Імовірність появи відмови підсистеми в разі появи відмовних ситуацій для діапазону кратності(m_r, m_q)

$$p_c^{(r,q)} = \sum_{r \le j \le q} \left(1 - \frac{N^{(j)}(t)}{N_0^{(j)}} \right) \cdot C_{n_j}^{m_j} \cdot \left(\sum_{j=0}^{n-n_j} C(n-n_j-k_j, k_j)\, p^{k_j} q^{n-n_j-k_j} \right) \cdot p^{m_j} q^{n_j-m_j}$$

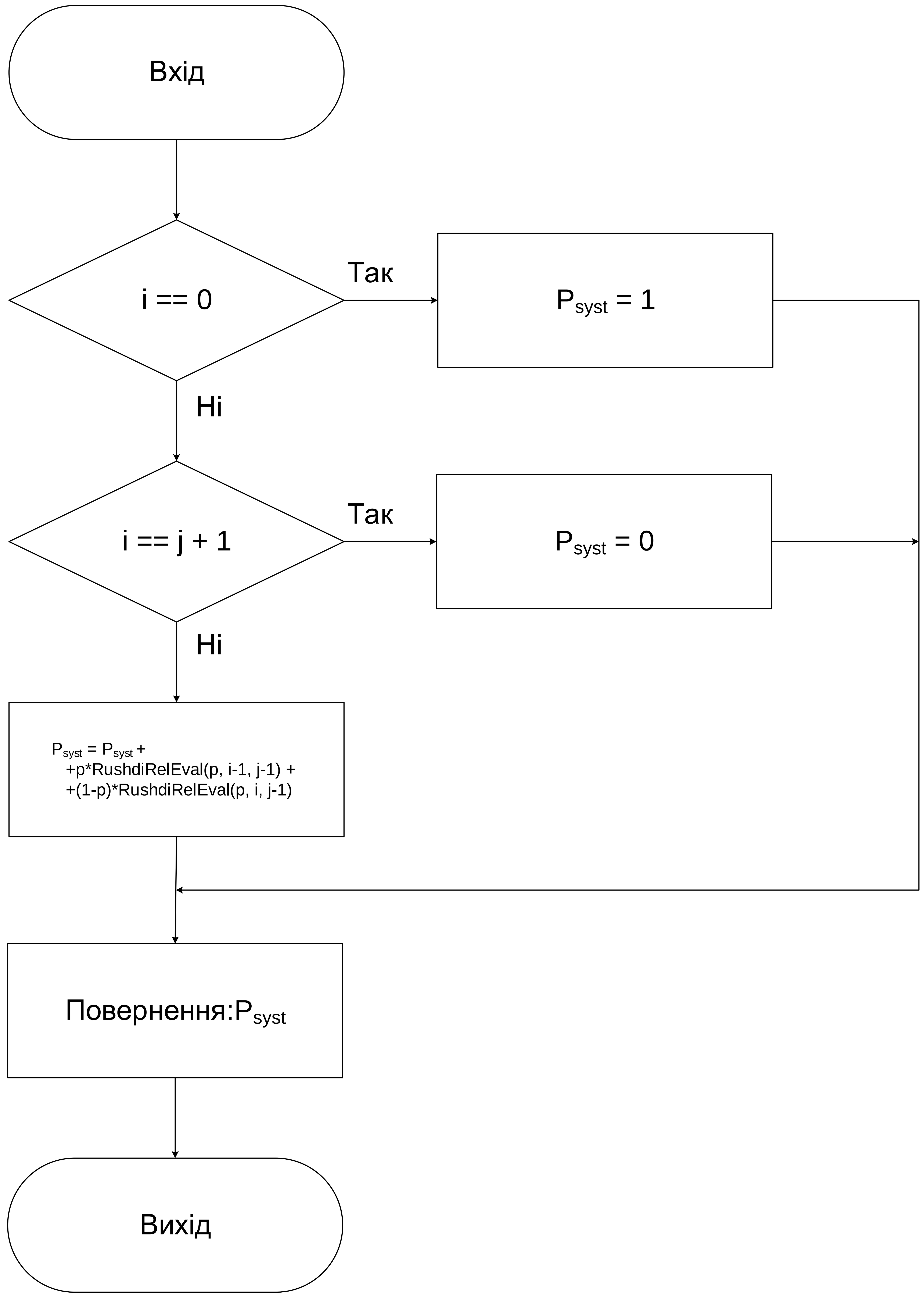
Корнєйчик Б.А. КВ-63М

АЛГОРИТМ РУШДІ

Основний алгоритм

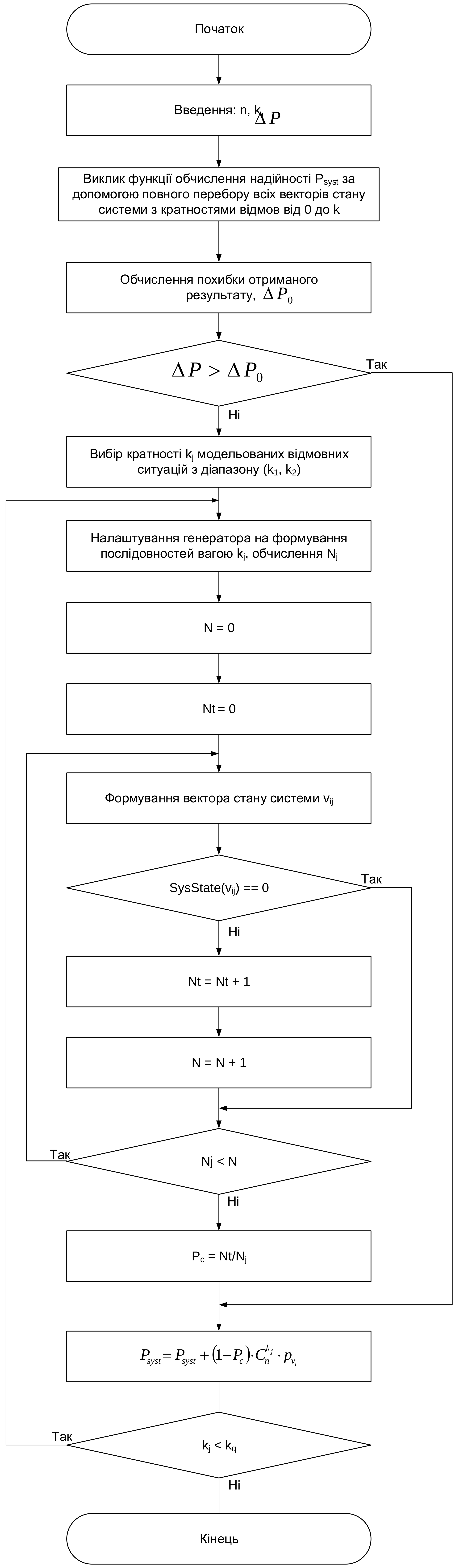


Рекурсивна функція обчислення надійності RusdiRelEval(p, n, k)



Корнєйчик Б.А. КВ-63М

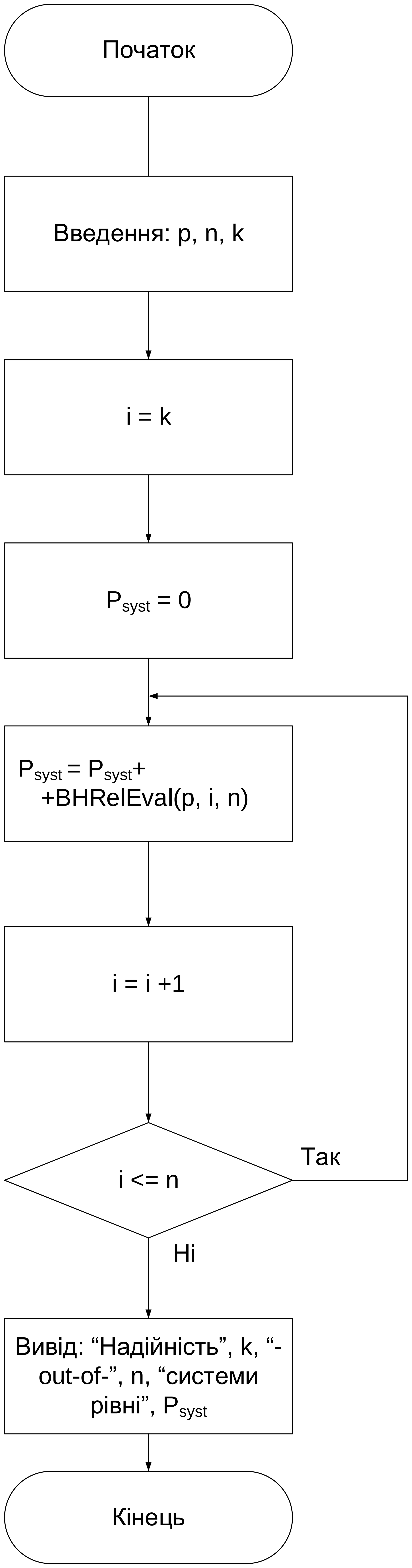
ЗАГАЛЬНИЙ АЛГОРИТМ РОЗРАХУНКУ НАДІЙНОСТІ СИСТЕМИ АБО ПІДСИСТЕМИ ЗА ДОПОМОГОЮ СТАТИСТИЧНОГО ЕКСПЕРИМЕНТУ З ГРАФО-ЛОГІЧНОЮ МОДЕЛЛЮ СИСТЕМИ



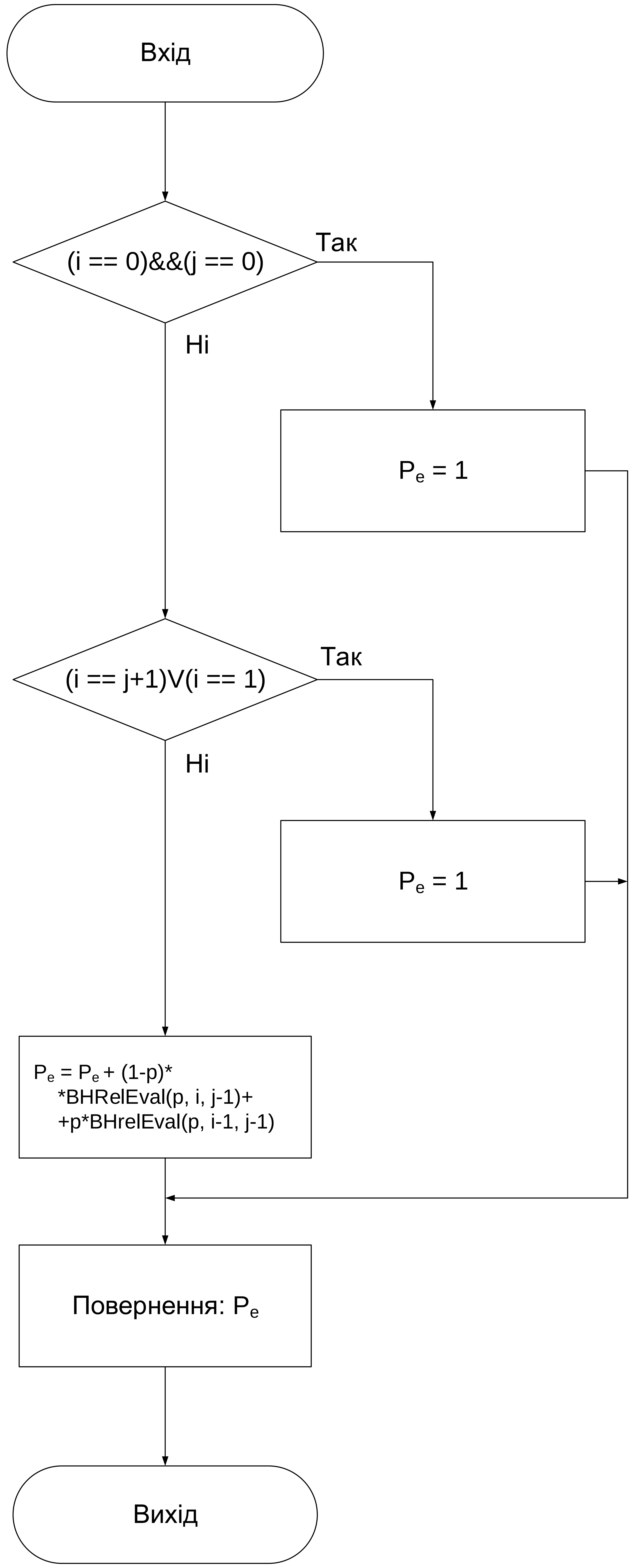
Корнєйчик Б.А. КВ-63М

АЛГОРИТМ БАРЛОУ ТА ХЕЙДТМАННА

Основний алгоритм

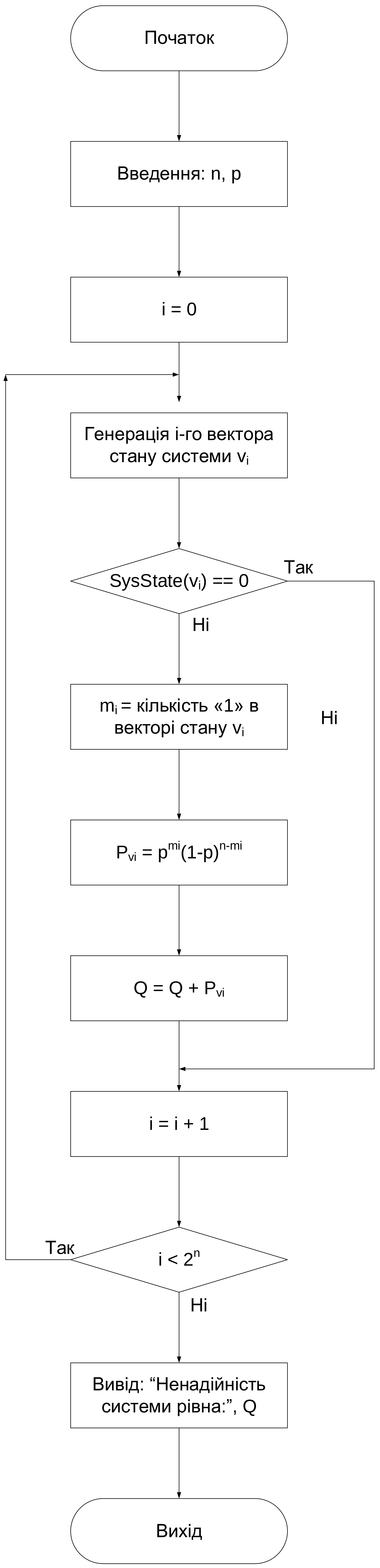


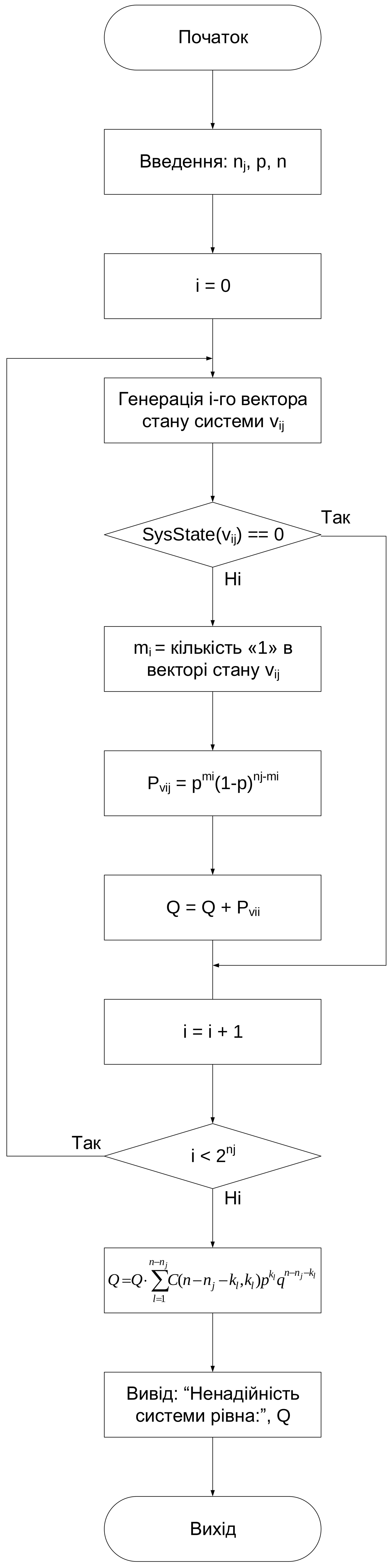
Рекурсивна функція обчислення надійності BHRelEval(p, k, n)



Корнєйчик Б.А. КВ-63М

ПОЧАТКОВИЙ ЕТАП РОЗРАХУНКУ НАДІЙНОСТІ (ФУНКЦІЯ ПОВНОГО ПЕРЕБОРУ ВІДМОВНИХ СИТУАЦІЙ) ДЛЯ СИСТЕМИ В ЦІЛОМУ І ДЛЯ ПІДСИСТЕМИ





Корнєйчик Б.А. КВ-63М

ФОРМУЛИ ДЛЯ РОЗРАХУНКУ НАДІЙНОСТІ ВБС СТАТИСТИЧНИМ МЕТОДОМ БЕЗ РОЗБИВКИ ВИХІДНОЇ СИСТЕМИ НА ПІДСИСТЕМИ

- Імовірність появи вектора стану системи, що складається з n елементів

$$p_{v_i} = p^{m_i} q^{n-m_i}$$

- Ненадійність і надійність системи, яка обчислюється при повному переборі векторів стану системи

$$Q_c = \sum_{v_{i_j}} \alpha(v_{i_j}) \cdot p_{v_i} = \sum_{v_i} \alpha(v_i) p^{m_i} q^{n-m_i}$$

$$P_c = 1 - \sum_{v_i} \alpha(v_i) \cdot p_{v_i} = 1 - \sum_{v_i} \alpha(v_i) p^{m_i} q^{n-m_i}$$

- Похибка розрахунку надійності системи, при заданій точності оцінки, яка повинна бути витримана

$$\Delta P_0 = \sum_{i=1}^{n-k} \frac{R(i) + C_n^{k+1}}{2} \cdot p^{k+i} q^{n-k-i}$$

- Імовірність появи відмови системи в разі впливу відмов модулів кратності m_j

$$p_c^{(j)} = \left(1 - \frac{N^{(j)}(t)}{N_0^{(j)}}\right) \cdot C_n^{m_j} \cdot p^{m_j} q^{n-m_j}$$

- Імовірність появи відмови системи в разі появи відмовних ситуацій для діапазону кратності (m_r, m_q)

$$p_c^{(r,q)} = \sum_{r \le j \le q} \left(1 - \frac{N^{(j)}(t)}{N_0^{(j)}}\right) \cdot C_n^{m_j} \cdot p^{m_j} q^{n-m_j}$$

Корнєйчик Б.А. КВ-63М

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Факультет прикладної математики

Кафедра системного програмування і спеціалізованих комп’ютерних систем

Д О В І Д К А   П Р О   В П Р О В А Д Ж Е Н Н Я

Видана Корнєйчику Борису Анатолійовичу у тому, що матеріали його магістерської дисертації «Метод обчислення надійності відмовостійкої багатопроцесорної системи, що складається з декількох підсистем» (керівник доц. Романкевич В.О.) впроваджені під час виконання д/б № 2808-ф «Методи оцінки та забезпечення необхідного рівня технічної безпеки роботи спеціалізованих багатопроцесорних систем управління».

Керівник теми
д.т.н., проф. Романкевич О.М.